

DEPARTMENT: Regular

A Multi-Tier Stacked Ensemble Algorithm for Improving Classification Accuracy

R. Pari
Research Scholar,

Dr M. Sandhya
Professor & Head

Dr Sharmila Sankar
Professor,

Department of Computer
Science & Engineering,

B. S. Abdur Rahman
Crescent Institute of Science
& Technology,
Chennai, India,

For real-world problems, ensemble learning performs better than the individual classifiers. This is true for **datasets which have many instances closer to the decision boundary**. Using a meta-learner to learn from the predictions of the base classifiers generalizes better. Hence, Stacked Ensemble (SE) is preferred over other ensemble methods. We extend SE and propose a **Multi-Tier Stacked Ensemble** (MTSE) algorithm with **three tiers namely a base tier, an ensemble tier and a generalization tier**. The base tier uses the traditional classifiers to predict the labels. **10-fold cross-validation is used to validate the models in the base tiers**. The cross-validated predictions are combined using combination schemes in the next tier. The predictions from the ensemble tier are generalized using meta-learning to give the final prediction. When tested with 36 datasets, MTSE gives superior performance over the stacked ensemble. It achieves high accuracy and is **not suffering from over-fitting/under-fitting**.

Ensemble learning combines the predictions of multiple weak classifiers to provide the final predictions¹⁻³. Ensemble learning improves the accuracy of classification, the robustness and the stability of the models. The ensemble performs better when the base models achieve more than 50% of accuracy and are diverse. Hence the

accuracy and diversity of classifiers dictate the effectiveness of ensemble learning. All the accurate models do not disagree with different parts of data, and hence there is a need to strike a perfect balance⁴. There are three major reasons why ensemble learning is better than individual classifiers. They are (i) statistical reasons (ii) computational reasons and (iii) representational reasons².

Rather than combining the results from the weak classifiers using a static function like average or weighted sum, trainable combiners generalize well. These ensemble systems are called as Stacked Ensemble (SE) or Stacked Generalization (SG)^{5,6}. Here, the combiner is also a classifier which is trained using the labels predicted by the weak classifiers⁷. The process of learning from the meta-knowledge produced by the weak classifiers is called as Meta-learning^{8,9}. In these systems, there are two levels of classifiers resulting in two levels of models. They are level-0 models and level-1 generalizer^{5,9}. The level-0 models are validated using leave-one-out k-fold cross-validation^{1,5,10,11}. The union of predictions for each of these k-folds along with the class labels in the original space gives the level-1 input space^{1,5}. Hence accuracy of SE is all about how well it can predict one part of the training set when taught with the rest of the training set. There are two major factors which determine the success of SE in improving the accuracy. They are (i) the type of features used to form level-1 space and (ii) the classifier used as level-1 generalizer. The class probabilities or the predictions of level-0 models are the most commonly used meta-features. Some of the previous works had used the entropy of class probabilities as meta-feature. Multiple Linear Regression (MLR) is the commonly used generalizer, time and again, it has been proved that any suitable classifier specific to the problem on hand can be used^{12,13}.

This study takes the stance that introducing another tier in-between the level-0 models and the level-1 generalizer in SE leads to better accuracy. The new tier combines the predictions from level-0 models and gets the intermediate predictions which serve as the meta-features. Using multiple combination schemes in the newly introduced tier leads to a better set of meta-features for level-1 generalizer. Training the generalizer on these meta-features gives accurate predictions. The contributions of this study are: (i) propose the innovative MTSE algorithm to improve the accuracy and (ii) encourage the analytics community to explore this algorithm for their problems. Organization of the remaining sections: Section II describes the related work in this area of the stacked ensemble. The methodology and the experimental evaluation of MTSE are depicted in Sections III and IV respectively. Section V is the conclusion and the scope for future work.

RELATED WORK

The research community has tried out many different extensions of SE. Alexander. K. Seewald et al.¹³ proposed Stacking C that used the class probability distribution of the base classifiers as the input for the regression models. He built one regression model for each of the classes in the input space. The output from these regression models was normalized to get the class probability distribution. For each instance of data, the classes with the highest probabilities were the predictions. The labels for the meta-learning were taken as either 1 or 0 depending upon whether that instance belongs to that particular class or not. In the case of binary classification problems, StackingC used only one linear regression model to predict the class labels. Using linear regression models as a generalizer is not suitable for problems where the conditional variance is not constant.

Developing an SE system to learn the brain images in a hierarchical fashion pays a rich dividend in classifying the brain related diseases. Manhua Liu et al.¹² developed an SE system to diagnose Alzheimer's disease. A hierarchical ensemble was built to combine the features and the decisions in a gradual manner. From the brain image, 'k' number of local patches were extracted. For each of these patches, the local imaging features and the correlation-context features were used for training two base classifiers. Thus 2*k number of base classifiers were trained. In addition to the predictions from the base classifiers, the coarse-scale imaging features were used to train the next level classifiers. The predictions from these classifiers were ensembled using weighted voting to give the final prediction. Training on the features of different brain regions helped in improving the classification. For the brain regions which were overlapping with each other, forward greedy search algorithm was used to select the classifiers to maximize the improvement of performance. The training set was divided into 10-fold for cross-validation. The frequencies of selection of the classifiers were used as the weights for the voting.

Florian Baumann et al.¹⁴ proposed a random forest framework which used a cascaded structure with several stages of decision trees to detect the objects. The complexity of the decision trees was increased from one stage to another stage. Weighted voting, full stage rejection and majority stage rejection along with the bootstrapping were implemented. In the first scheme, the predictions from different stages were combined using the weighted majority voting. The stages with low accuracy were assigned lower weights based on their F1 scores. In the second scheme, the objects were accepted if all the stages passed them. In the third scheme, the images were accepted if more than half of the stages passed them. In every stage, true negatives were removed and re-filled with false positives. When tested with pedestrian detection, car detection and unconstrained face detection datasets, the objects were detected with less time as compared to the non-cascaded trees.

Asif Ekbal et al.¹⁵ used the conditional random field (CRF) and SVM as the level-0 classifiers. A diverse set of features like context words, word prefix & suffix, word length, infrequent word, part-of-speech information, chunk information, dynamic feature, unknown token feature, word normalization, head nouns, verb trigger, word class feature, informative words, content words in surrounding contexts and orthographic features were used to train the level-0 models. A genetic algorithm was used to select the features and to optimize the level-0 models. For each of the set of features produced by GA, CRF and SVM models were trained. The predictions from these models along with the features selected by GA formed the input space for level-1 generalizer. CRF was used as level-1 generalizer. Recall, precision and F-measure were used to evaluate the models. When tested on the JNLPBA dataset and GENETAG dataset, the F-measure was found to be 75.17% and 94.70% respectively.

METHODOLOGY

The major challenge with SE is that either the bias or the variance of level-0 models gets cascaded to level-1 generalizer. Due to this, there is still some level of over-fitting or under-fitting. It affects the performance of the ensemble system. To overcome the problem of over-fitting or under-fitting, there is a need to fine-tune SE so that it achieves better performance. In the literature, many ways of fine-tuning SE are discussed. Some of them focus on the features used in the level-1 space and others focus on the type of generalizer used in level-1⁷. Though these approaches have resulted in improving the performance, they have not reached the desired level of performance. Hence there is a scope for further improvement in performance. In this study, SE is extended by introducing another tier between the level-0 classifiers and the level-1 generalizer. The proposed algorithm is named MTSE.

As depicted in figure 1, MTSE has three tiers. The base tier uses a set of weak classifiers which are suitable for the problem at hand. The ensemble tier uses the combination schemes to produce the combined predictions based on the results of the weak classifiers. The generalization tier does the Meta-learning based on the predictions obtained from multiple ensemble classifiers and the class labels in the known samples. This helps to control the bias and the variance in two different tiers. The base tier takes care of reducing the bias, and the other two tiers take care of reducing the variance. This leads to a perfect tradeoff between bias and variance and hence results in the improvement in performance. The base tier in MTSE uses the following classifiers.

- Support Vector Machines (SVM)
- Decision Trees (DT)
- Logistic Regression (LR)
- K-Nearest Neighbors (kNN)
- Gaussian Naïve Bayes (GNB)
- Stochastic Gradient Descent (SGD)
- Passive Aggressive Classifier (PAC)
- Perceptron (Linear Model)

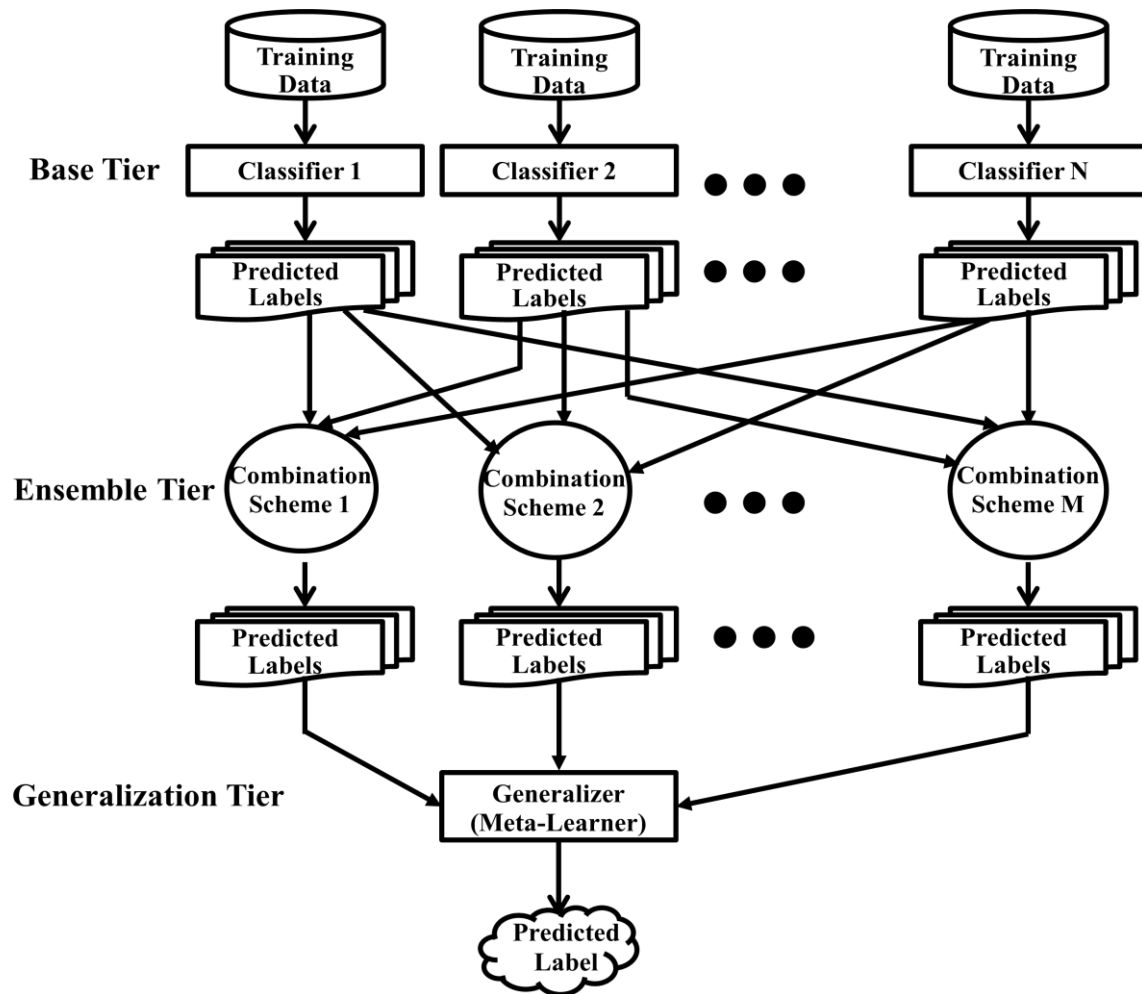


Figure 1. Multi-Tier Stacked Ensemble Algorithm. Depicts the data processing at different tiers of MTSE.

The dataset is split into two sets S1 and S2 at 80% and 20% proportionately. S1 is used for training and S2 is used for the final testing. In the base tier, 10-fold cross-validation is used to reduce the bias. The set S1 is partitioned into ten sets of equal size. In each fold, nine of these sets combined are used for training the base models, and the left-out set is used for testing the base models. This is repeated for ten folds by selecting a different set for testing. Thus each set is used nine times for training and one time for testing. This results in the cross-validated predictions for all the instances in S1. The ensemble tier uses the following combination schemes.

- Plural Voting
- Majority Voting
- Weighted Majority Voting
- Confidence-Based Voting

Logistic Regression is used as the meta-classifier in the generalization tier. The critical features from the input space are selected based on their co-variance with the class labels. The top three features with high co-variance values are selected. The predictions from the combination schemes along with the selected features form the input space for generalization tier. Both SE and MTSE are tested with the set S2. Based on this the accuracies of the models are compared.

The following are the two reasons why MTSE performs better than SE:

- Consider the predictions from each of the base classifiers as a random variable. The combination scheme in the ensemble tier is a joint distribution of these random variables. The joint distribution takes all the pair-wise interactions of the random variables into consideration. The multiple combination schemes define multiple joint distributions of the random variables. Each of these random variables is also a function which maps the input space to the class labels. Due to this, the output of the combination schemes provides features which not only encapsulate the relationship between the predictions from the base tier and the actual labels but also the relationship between the

input space and the class labels. These features provide more quality information about the input space than the original features in the input space. Hence training a model on these features yields models with better accuracy.

- Depending upon the misclassification rates of the base classifiers, the predictions from each of the base classifier contains a set of False Positives (FP) and False Negatives (FN). When appropriate classifiers are used in the base tier, all these sets become more or less complement to each other. With a larger pool of base classifiers, the number of common elements between them tends to zero. When the base classifiers disagree for different sets of examples, the combination schemes in the multi-tier ensure that the weaknesses of the base classifier are neutralized with each other. Hence the combination schemes produce better features. When these features are trained using a suitable classifier, it generalizes better than SE.

In addition to the above-stated reasons, MTSE is suitable for most of the classification problems due to the following reasons also.

- Statistical Reason: The ensemble tier in MTSE combines the predictions of the base classifiers using different combination schemes. This ensures that the **risk of choosing a wrong classifier is reduced**. The risk is further reduced at the generalization tier, as it generalizes the outputs from multiple combination schemes.
- Computational Reason: MTSE completely **eliminates the possibility of finding the local optima**. The diversity in MTSE ensures the better approximation of the underlying function between the input data and the class label.
- Representational Reason: Even if each of the base classifiers cannot truly represent the underlying function, the combination schemes ensure that the space of the underlying function is expanded. The generalization tier expands this space further. Hence MTSE can truly represent the underlying function.

Mathematical Model

Let $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ be the dataset, where $x_i \in X, y_i \in Y = \{c_1, c_2, \dots, c_l\}$ and N is the total number of instances in the dataset and c_1, c_2, \dots, c_l are the class labels. Typically the input space X consists of many features and hence its elements are represented as N -tuple of 'd' dimensions.

$$X = \{(x_{11}, x_{12}, \dots, x_{1d}), (x_{21}, x_{22}, \dots, x_{2d}), \dots, (x_{N1}, x_{N2}, \dots, x_{Nd})\}. \quad (1)$$

where 'd' is the number of features in the input space. Hence, the dataset is represented as:

$$D = \{((x_{11}, x_{12}, \dots, x_{1d}), y_1), ((x_{21}, x_{22}, \dots, x_{2d}), y_2), \dots, ((x_{N1}, x_{N2}, \dots, x_{Nd}), y_N)\}. \quad (2)$$

Split D into two sets S_1 and S_2 at 80% and 20% respectively. S_1 is used as a training set and S_2 is used as a test set.

$$D = S_1 \cup S_2. \quad (3)$$

Base Tier

Let $WC = \{WC_1, WC_2, \dots, WC_m\}$ be the set of weak classifiers.

Let T_1, T_2, \dots, T_{10} be the equal size sets, partitioned from the set S_1 .

Fold 1: The training set $T = T_1 \cup T_2 \cup \dots \cup T_9$

The testing set $V = T_{10}$

Train the weak classifiers using T .

The weak classifiers WC_m maps the elements of T into one of the classes in Y .

$$WC_k: X_g \rightarrow Y \quad (4)$$

$$WC_k: X_g \rightarrow \{c_1, c_2, \dots, c_l\} \quad (5)$$

$$\text{where } k=1, 2, \dots, m \text{ and } X_g \text{ belongs to } T \quad \text{i.e. } X_g \in T \quad (6)$$

In the next fold, T_9 is taken as the testing set and the remaining nine sets combined together are taken as the training set. This is repeated until all the partitioned sets in S_1 are taken as testing sets. This results in the predictions for all the instances in T . Let $WL = \{WL_1, WL_2, \dots, WL_m\}$ be the set of labels predicted by the weak classifiers.

$$\text{i.e. } WL_k = WC_k(X_g) \quad (7)$$

where $k = 1, 2, \dots, m$

Ensemble Tier

Let $EC = \{EC_1, EC_2, \dots, EC_m\}$ be the set of combination schemes and they combine the labels predicted by the weak classifiers and map them into one of the classes in Y .

$$EC_k: WL_k \rightarrow Y \quad (8)$$

$$EC_k: WL_k \rightarrow \{c_1, c_2, \dots, c_l\}. \quad (9)$$

$$EC_k = \text{Aggregation of } (WL_k) \quad (10)$$

$$EC_k = \text{Aggregation of } (WC_k(X_g)). \quad (11)$$

where $k = 1, 2, \dots, m$

Let $EL = \{Y_{ec1}, Y_{ec2}, \dots, Y_{ecm}\}$ be the set of labels predicted by the ensemble classifiers. This can be represented in mathematical form as

$$Y_{eck} = EC_k(WL_k), \quad (12)$$

$$Y_{eck} = EC_k(WC_k(X_g)). \quad (13)$$

where $k = 1, 2, \dots, m$

Generalization Tier

The generalization tier takes the top three critical features $CF = \{CF_1, CF_2, CF_3\}$, ensemble predictions EL and the actual labels Y as input. It produces the generalized predictions. This can be mathematically represented as

Input : $\{CF, EL, Y\}$ or $\{\{CF_1, CF_2, CF_3\}, EL, Y\}$

$$GL: \{CF, EL\} \rightarrow Y \quad (14)$$

$$GL: \{CF, EL\} \rightarrow \{c_1, c_2, \dots, c_l\} \quad (15)$$

$$GL = GC(\{CF, EL\}) \quad (16)$$

$$GL = GC(\{CF, EC_k(WL_k)\}) \quad (17)$$

$$GL = GC(\{CF, EC_k(WC_k(X_g))\}). \quad (18)$$

where $k = 1, 2, \dots, m$, GC is the generalization classifier or meta-learner and GL is the final prediction produced by the meta-learner. GC generalizes the labels predicted by the ensemble classifiers into one of the classes in Y .

Accuracy of Classification

In this study, the accuracy of the classifiers is calculated using the test set S_2 . The ratio between the count of correctly classified instances to the total number of instances classified gives the accuracy.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)}, \quad (19)$$

where TP is #True Positives, TN is #True Negatives, FP is #False Positives and FN is #False Negatives in the final prediction. As MTSE focuses on improving the TP and the TN , accuracy is the only performance measure considered in this study.

Algorithm

Algorithm 1: Classification by Base Classifiers

- 1: **Input:** Dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
 $\{ \mid x_i \in X, y_i \in \{c_1, c_2, \dots, c_l\} \}$
- 2: **Output:** The labels (WL) predicted by the weak classifiers
- 3: **Randomly split** D into two sets of size 80% and 20% and name them as training set S_1 and test set S_2 .
- 4: **Randomly split** S_1 into ten sets of equal size 10% and name them as T_1, T_2, \dots, T_{10}
- 5: **Do for** $m=1, 2, \dots, M$: //Repeat for eight weak classifiers


```

6:   Do for i = 1, 2, ..., K:   //Repeat for each of the ten folds
7:        $T \leftarrow \bigcup_{j=1}^{j=10} T_j$ , except for  $i == j$ 
8:        $V \leftarrow T_i$ 
9:       Train a weak classifier  $WC_m$  with  $T$ 
10:      Run the weak classifier  $WC_m$  with  $V$  and get the predictions  $WL_{mi}$ 
11:      End
12:      Combine the predictions of all the ten folds of the weak classifier  $WC_m$  into a
single set.  $WL_m = WL_{m1} \cup WL_{m2} \cup \dots \cup WL_{m10}$ 
13:      Run the weak classifier  $WC_m$  with  $S_2$  and get the predictions  $WLT_m$ 
14:      Calculate the Accuracy of the weak classifier  $WC_m$ :

$$A_m \leftarrow \frac{(TP_m + TN_m)}{(TP_m + TN_m + FP_m + FN_m)}$$

15:  End
16:  End

```

Algorithm 2: Classification by Combination Schemes

```

1: Input: The labels (WL) predicted by the weak classifiers
2: Output: The labels (EL) predicted by the ensemble classifiers
3: Plural Voting:
4: Do for n = 1, 2, ..., N:
 $EL_n \leftarrow \text{Mode of } (WL_{1n}, WL_{2n} \dots WL_{mn})$ 
5: End
6: Simple Majority Voting:
7: Do for n = 1 to N
8:   Corresponding to each class label, initialize a counter to zero.
9:   Do for m = 1 to 8
10:    For each occurrence of a class label in the classifier's
prediction, increment the corresponding counter
11:   End
12:    $Max\_Count \leftarrow \text{Max}(\text{All Counters})$ 
13:   If  $Max\_Count > 4$ :
14:      $EL_n \leftarrow$  The class label corresponding to the  $Max\_Count$ 
15:   Endif
16: Confidence-Based Voting:
17: For all the instances, store the class probabilities of all the classifiers for all the
classes
into an array
18: Do for n = 1, 2, ..., N:
19:    $Max\_Prob \leftarrow \text{Max}(\text{class probabilities of all the classifiers for all the classes})$ 
20:    $EL_n \leftarrow$  The class label corresponding to  $Max\_Prob$ 
21: End
22: Weighted Majority Voting:
23: Use the Brute Force Algorithm to find the optimal weights of the base learners.
 $W = \{W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8\}$ 
24:  $WTot \leftarrow W_1 + W_2 + W_3 + W_4 + W_5 + W_6 + W_7 + W_8$ 
25: Do for n = 1, 2, ..., N:
26:   Do for k = 1, 2, ..., L:   // Repeat for each class label
27:      $WSum_{nk} \leftarrow 0$ 
28:     Do for m = 1, 2, ..., M //Repeat for each weak classifier
//Find the sum of the products of the weights ( $W_m$ )
and the class probabilities ( $P_{mk}$ )
29:      $WSum_{nk} \leftarrow WSum_{nk} + W_m * P_{mk}$ 
30:     End
31:   End
32:    $WAvg \leftarrow \frac{WSum_{nk}}{WTot}$ 
32:    $EL \leftarrow$  Class label with highest weighted average
33: End
35: Calculate the Accuracy of the ensemble classifier
36: End

```

Algorithm 3: Classification by Meta-learner

```

1: Input: The labels (EL) predicted by the ensemble classifiers
2: Output: The final labels (GL) predicted by the generalizer
3: Calculate the covariance with a class label for each of the features in the input space
   // Used for picking the critical features
4:  $CF \leftarrow$  Pick three features with high covariance values
5: Combine the critical features and the ensemble predictions
    $TS \leftarrow CF \cup EL$ 
6: Train the generalizer with the set TS
7: Run the model obtained in step 6 with the set  $S_2$ 
8: Calculate the accuracy of the generalizer
9: End

```

Figure 2 depicts the input and the output of each of these algorithms for a simple example with three instances with three features. To keep it simple, this figure only shows how the predictions are made for the unseen samples. Training these algorithms also follows the similar processing. Here only three classifiers are considered in the base tier, and three combination schemes are considered in the ensemble tier. In this example, only the top two critical features are used for meta-learning. This example demonstrates how the labels predicted by the weak classifiers are combined as a matrix and is used as input for the ensemble tier. Similarly, it shows how the combined output from the combination schemes and the top two features together are combined as a matrix and used as input for meta-learning.


Algorithm	Input	Output	Description
Algorithm 1:	$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{21} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$	$WL = \begin{bmatrix} c_2 & c_3 & c_2 \\ c_1 & c_1 & c_1 \\ c_3 & c_3 & c_2 \end{bmatrix}$	Where each column represents the labels predicted by each of the weak classifiers i.e. $WL_1 = \begin{bmatrix} c_2 \\ c_1 \\ c_3 \end{bmatrix}$, $WL_2 = \begin{bmatrix} c_3 \\ c_1 \\ c_3 \end{bmatrix}$ and $WL_3 = \begin{bmatrix} c_2 \\ c_1 \\ c_2 \end{bmatrix}$
Algorithm 2:	$WL = \begin{bmatrix} c_2 & c_3 & c_2 \\ c_1 & c_1 & c_1 \\ c_3 & c_3 & c_2 \end{bmatrix}$	$EL = \begin{bmatrix} c_3 & c_3 & c_2 \\ c_1 & c_1 & c_1 \\ c_2 & c_3 & c_2 \end{bmatrix}$	Combination Scheme 1: $\sum \begin{bmatrix} c_2 & c_3 & c_2 \\ c_1 & c_1 & c_1 \\ c_3 & c_3 & c_2 \end{bmatrix}$ gives $EL_1 = \begin{bmatrix} c_3 \\ c_1 \\ c_2 \end{bmatrix}$ as the output. Similarly other combination schemes give $EL_2 = \begin{bmatrix} c_3 \\ c_1 \\ c_2 \end{bmatrix}$ and $EL_3 = \begin{bmatrix} c_3 \\ c_1 \\ c_2 \end{bmatrix}$ as the output.
Algorithm 3:	$EL + CF = \begin{bmatrix} c_3 & c_3 & c_2 & x_{11} & x_{31} \\ c_1 & c_1 & c_1 & x_{12} & x_{32} \\ c_2 & c_3 & c_2 & x_{13} & x_{33} \end{bmatrix}$	$GL = \begin{bmatrix} c_3 \\ c_1 \\ c_2 \end{bmatrix}$	Where, CF is the set of top two features from the input space The Generalizer takes the combined matrix of EL and CF and produces the final predictions

Figure 2. A simple example to demonstrate how the prediction is made using MTSE. Demonstrates the input and output for each of the algorithms.

Experimental Evaluation

The experiment was implemented using Python with Scikit-learn library. **Synthetic Minority Over-sampling Technique (SMOTE)** was used for balancing the classes. The experiment was conducted with the datasets listed in Table 1. These datasets are publicly available in the UCI Machine Learning repository. One-vs.-rest and group-vs.-rest strategies were used for grouping the classes and to derive the data sets. Thus 36 datasets were derived.

Table 1. Summary of Datasets used in the Experiment

Datasets	# instances	# attributes	# classes
Ecoli	336	7	8
Abalone	4177	8	29
Yeast	1484	8	10
Glass	214	9	7
Shuttle	58000	9	7
Page Blocks	5473	10	5
Heart 	270	13	2
Wine	178	13	3
Segment	210	19	7
Thyroid-ann	3772	21	3
Thyroid-allrep	1947	28	4
Ionosphere	351	34	2

For the purpose of benchmarking, the traditional SE, Rotation Forest, AdaBoost and Extremely Randomized Trees were used. The comparison of the mean accuracy of MTSE and the other four algorithms is depicted in Table II. MTSE has achieved a significant improvement in accuracy for almost all the datasets. Though other algorithms have reached the accuracy of 100% for some of the datasets, MTSE is not dragging it down, and it also reaches 100% for those datasets. Thyroid-allrep4 is the only dataset for which the performance of MTSE (99.57%) is little less than that of SE (99.78%). This is a reduction of 0.21% concerning the performance of SE and is not a considerable reduction.

Table 2. Comparison of Accuracies of MTSE and Other Algorithms

Dataset	MTSE	SE	RF	AB	ERT
Ecoli0v1	99.39	98.61	97.55	98.03	97.96
Ecoli3	97.22	95.77	93.42	93.66	94.88
Ecoli4	98.91	96.02	93.97	95.71	95.71
Ecoli678	100.0	99.39	94.41	98.17	98.17
Abalone5v19	100.0	99.44	98.29	99.44	99.39
Abalone7v17	99.11	97.34	97.83	97.34	97.06
Abalone9v18	95.4	94.78	92.96	93.04	94.27
Abalone19	98.84	98.6	95.87	95.51	95.58
Yeast1v3	96.29	94.41	95.69	94.41	95.91
Yeast0v4	100.0	99.15	99.36	99.15	99.28
Yeast1v7	96.29	95.81	96.13	92.55	94.76
Yeast6	98.9	98.34	98.82	96.55	98.19

Dataset	MTSE	SE	RF	AB	ERT
Glass2	94.44	89.85	84.33	85.50	88.46
Glass1	94.44	90.27	86.71	87.50	89.39
Glass7	100.0	99.02	84.89	99.46	99.52
Glass5	100.0	99.5	84.68	99.00	98.36
Glass6	100.0	99.51	84.12	99.00	98.79
Shuttle5v3	100.0	100.0	98.72	100.0	100.0
Shuttle4v2	100.0	100.0	99.01	100.0	100.0
Page Blocks2v4	98.78	98.18	98.43	97.88	96.32
Page Blocks45v3	100.0	100.0	99.89	99.01	99.23
Page Blocks25v3	100.0	100.0	99.76	99.54	99.54
Page Blocks1v5	99.34	99.3	99.18	98.81	99.07
Page Blocks1v3	100.0	99.91	99.47	99.83	99.61
Heart1v2	100.0	100.0	99.49	100.0	100.0
Wine2	100.0	99.03	99.51	98.10	100.0
Wine1	100.0	99.16	99.38	99.16	100.0
Wine3	100.0	99.16	99.43	99.16	100.0
Segment123	92.85	92.45	92.03	88.79	91.96
Segment1	99.36	99.01	98.65	98.88	98.88
Thyroid-ann2	100.0	100.0	99.03	100.0	100.0
Thyroid-ann1	100.0	100.0	98.97	100.0	100.0
Thyroid-allrep23v1	95.00	92.00	94.09	93.12	94.72
Thyroid-allrep4	99.57	99.78	99.33	99.71	99.71
Thyroid-allrep3	99.89	99.89	99.29	99.72	99.84
Ionosphere	97.34	95.57	93.88	94.69	94.69

The impact of introducing a new tier into SE was studied across the datasets to find out the number of datasets for which (i) the accuracy improved (ii) the accuracy remained the same and (iii) the accuracy went down. This analysis is depicted in figure 3. Overall, the performance of MTSE is superior to other algorithms for all the datasets. Amongst the 36 datasets used for experimenting, there are a total of 35 datasets, where MTSE is either better than other algorithms or on par with them. MTSE has achieved a maximum improvement of 4.59% for Glass2 dataset. Glass1, Allrep23v1 and Ecoli4 take the next three places regarding the improvement in accuracy. For these three datasets, MTSE has achieved 4.17%, 3% and 2.89% of improvement over SE. Only for the Thyroid-allrep4 dataset, the introduction of the new tier has marginally reduced the accuracy. From 99.78%, it has gone a tad down to 99.57%, a reduction of 0.21% in accuracy. This reduction is further analyzed to find the fact that the features in this dataset are not strongly correlated with the class labels. Thus, an important caveat of MTSE is that it requires the features of the class labels to have a strong correlation with the class labels. This caveat sets the direction for the future work for this study. Exploring the use of Principal Component Analysis (PCA) to select the principal components for each of the datasets and then classifying the

datasets is an interesting area of research. Using only the principal components rather than using all the features improves both the training time and the prediction time.

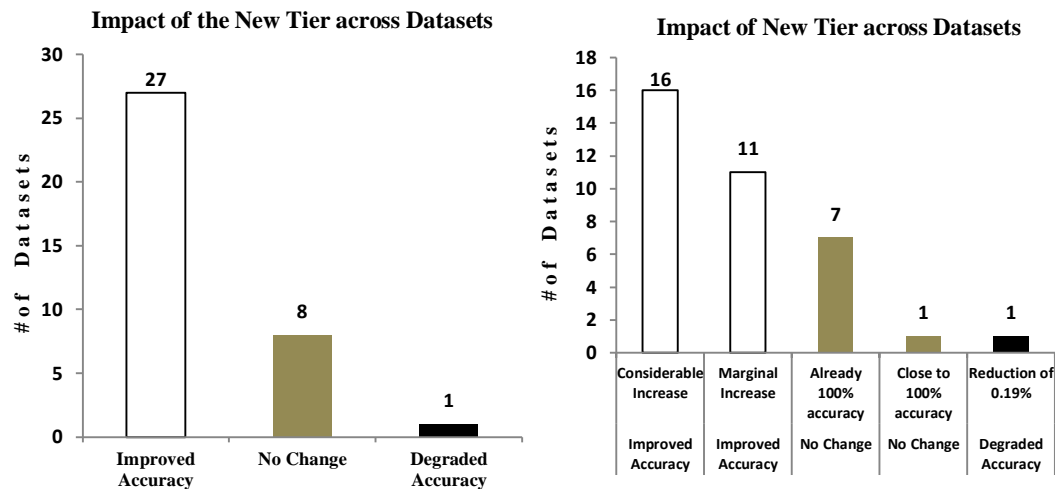


Figure 3. Impact of Introducing a New Tier into SE on Accuracy. Due to the introduction of the new tier, for 27 datasets, the accuracy had improved. For eight datasets, there was no change in the accuracy and for one dataset, the accuracy has gone down.

Analyzing the performance of MTSE separately without comparing it with SE gives useful inferences. Figure 4 depicts this analysis. MTSE has reached the accuracy of 100% for 17 datasets. MTSE's accuracy is greater than 98% for 28 datasets. MTSE's accuracy is between 96 and 98% for four datasets and is between 94 and 96% for three datasets. Segmentation123 is the only dataset where MTSE has achieved less than 94% accuracy. MTSE's accuracy for this dataset is 92.85% which is still greater than the accuracies of other algorithms. On the whole, for 35 datasets, MTSE's accuracy is greater than 94%. Once again this establishes the fact that MTSE gives superior performance for a wide range of datasets. To understand the rationale behind MTSE's performance of greater than 98% of accuracy for the 28 datasets, the correlation between the features was studied. It reveals that the features in these 28 datasets are having less correlation between themselves. Hence these datasets have features which are almost independent of each other. On the other hand, the other eight datasets have some of the features which are inter-related. This inference leads the way for future work of this research. Carrying out Feature Engineering using Principal Component Analysis improves the performance.

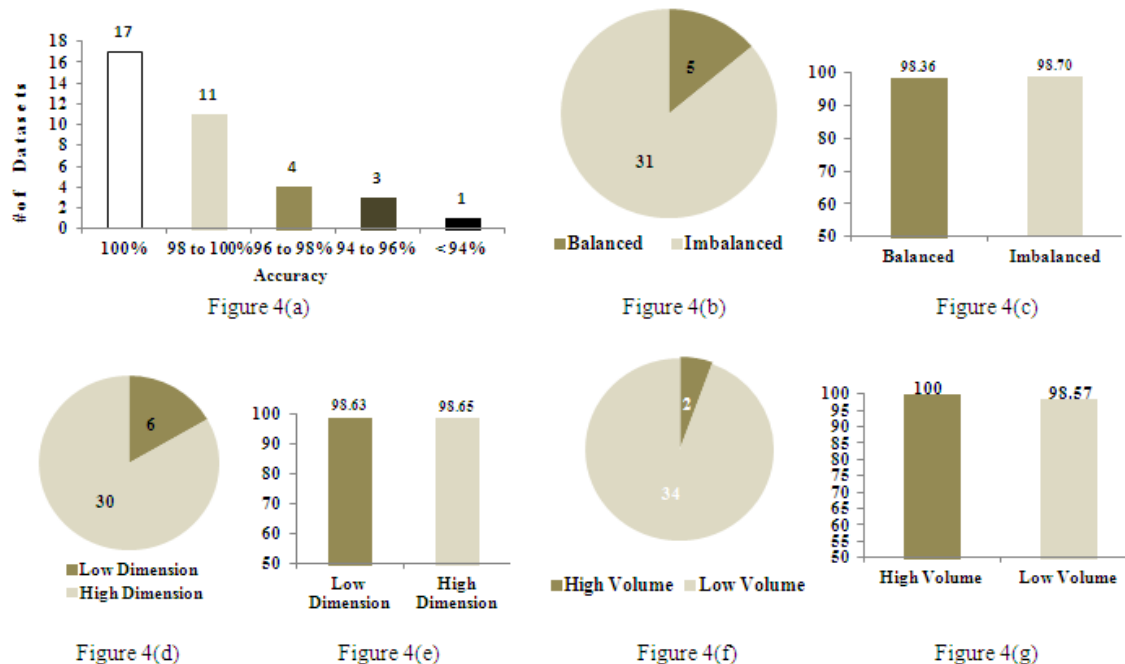


Figure 4. Performance of MTSE across the Datasets. 4(a). Accuracy distribution of MTSE across the datasets. 4(b). Count of Balanced and Imbalanced Datasets. 4(c). Average accuracies of all the balanced datasets and the imbalanced datasets. 4(d). Count of High and Low Dimension datasets. 4(e) Average accuracies of High and Low Dimension datasets. 4(f). Count of High and Low Volume datasets. 4(g). Average accuracies of High and Low Volume datasets

MTSE performed well for both balanced and imbalanced datasets. In the case of unbalanced datasets, MTSE's performance was measured after the dataset was balanced using SMOTE. The performance of MTSE is marginally better for imbalanced datasets. Hence using SMOTE for balancing the datasets has a positive impact on the performance of MTSE. Another reason for the marginal increase in the accuracy is that the number of training examples increases when the datasets are balanced using SMOTE. For this analysis, datasets with an Imbalance Ratio (IR) of greater than two are considered as imbalanced datasets. Out of the 36 datasets, 31 datasets have IR greater than two, and five datasets have IR less than or equal to two.

There is no significant change in the performance of MTSE between high and low dimensional space. For this analysis, datasets with more than 20 features are considered as high dimensional space. Out of the 36 datasets, 30 datasets have more than 20 features, and six datasets are having less than or equal to 20 features. The dimension of the dataset impacts only the base tier. Beyond which the number of features in the input space is dependent on the number of base classifiers and the number of combination schemes. The performance of MTSE is much better for high volume dataset when compared to the low volume datasets. The volume of the dataset impacts all the three tiers and hence a proportionate increase in the accuracy. For this analysis, datasets with more than 50,000 instances are considered as high volume datasets. Out of the 36 datasets, Shuttle5v3 and Shuttle4v2 are the only datasets with 58,000 instances. Hence there are the two high volume datasets, and the remaining 34 are low volume datasets. For the two high volume datasets, MTSE has achieved a performance of 100%. For the low volume datasets, the average performance is 98.57%. Another interesting inference is that these two high volume datasets are having an IR of 18.62 and 182.38 respectively. Due to the high value of IR, SMOTE also has played a vital role in improving the performance of MTSE for these two datasets.

For the high volume dataset (Shuttle dataset with 58,000 instances), the execution time of MTSE and other algorithms are:

- MTSE – 17.48 seconds
- SE – 16.81 seconds
- RF – 13.53 seconds
- AB – 12.77 seconds
- ERT – 11.96 seconds

Due to the additional data processing happening in the newly introduced tier, the execution time of MTSE is higher than the other algorithms. Considering the improvement in accuracy and the computing resources available in the current scenario, the increase in the execution time of MTSE is not consequential. The parallel processing of the base learners and the parallel processing of the combination schemes in the ensemble tier can be considered for the future work.

CONCLUSION

This study introduced an MTSE algorithm by adding a new tier into the traditional SE. The experimental results point out that MTSE achieved a better accuracy than SE. From this study, it is evident that the introduction of a new tier into SE gives superior performance for balanced/unbalanced datasets, high/low dimension datasets and high/low volume datasets. Hence we recommend the use of MTSE for the real-time classification problems in any domain. This study can further be enhanced by carrying out feature engineering for the datasets. Feature engineering is expected to improve the performance further and hence make MTSE the most reliable solution for classification problems. In addition to this, we are also exploring if MTSE is suitable for incremental learning using streaming data.

REFERENCES

1. Polikar, R. Ensemble based systems in decision making. *Circuits Syst. Mag. IEEE* **6**, 21–45 (2006).
2. Dietterich, T. G. Ensemble Methods in Machine Learning. *Mult. Classif. Syst.* **1857**, 1–15 (2000).

3. Kittler, J., Hater, M. & Duin, R. P. W. Combining classifiers. *Proc. - Int. Conf. Pattern Recognit.* **2**, 897–901 (1996).
4. Krogh, A. & Vedelsby, J. Neural Network Ensembles, Cross Validation, and Active Learning. 231–238 (1994).
5. Wolpert, D. H. Stacked Generalization. *Neural Networks* **5**, 241–259 (1992).
6. Opitz, D. W. & Maclin, R. Popular Ensemble Methods: An Empirical Study. *J. Artif. Intell. Res.* **11**, 169–198 (1999).
7. Ting, K. M. & Witten, I. H. Issues in stacked generalization. *J. Artif. Intell. Res.* **10**, 271–289 (1999).
8. Breiman, L. Stacked regressions. *Mach. Learn.* **24**, 49–64 (1996).
9. Vilalta, R. & Drissi, Y. A perspective view and survey of meta-learning. *Artif. Intell. Rev.* **18**, 77–95 (2002).
10. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Appear. Int. Jt. Conf. Artificial Intell.* **5**, 1–7 (1995).
11. Stone, M. Cross-validated choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B* 111–147 (1974).
12. Liu, M., Zhang, D., Yap, P. T. & Shen, D. Hierarchical ensemble of multi-level classifiers for diagnosis of Alzheimer’s disease. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **7588 LNCS**, 27–35 (2012).
13. Seewald, A. *How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness. Proceedings of the 19th International Conference on Machine Learning* (2002).
14. Baumann, F., Ehlers, A., Vogt, K. & Rosenhahn, B. Cascaded random forest for fast object detection. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **7944 LNCS**, 131–142 (2013).
15. Ekbal, A. & Saha, S. Stacked ensemble coupled with feature selection for biomedical entity extraction. *Knowledge-Based Syst.* **46**, 22–32 (2013).

ABOUT THE AUTHORS

R. Pari received M. Tech. Degree from PRIST University, India. He is a research scholar in the Department of Computer Science and Engineering at B.S. Abdur Rahman Crescent Institute of Science and Technology, India. His research interests include Machine Learning, Artificial Intelligence and Software Engineering.

Dr M. Sandhya is a Professor and Head, Department of Computer science and Engineering at B. S. Abdur Rahman Crescent Institute of Science and Technology, India. She contributed to the research and education in Wireless Networks, Cloud Computing and Big Data Analytics. She is a life member in Indian Society for Technical Education. She is a reviewer for many international journals.

Dr Sharmila Sankar is a Professor of Computer Science and Engineering at B. S. Abdur Rahman Crescent Institute of Science and Technology, India. She contributed to the research and education in Internet of Things, Wireless Networks and Big Data Analytics. She is a member of Association for Computing Machinery and Computer Society of India. She is a reviewer for many international journals.