Improvement in Classification Algorithms through Model Stacking with the Consideration of their Correlation

Muhammad Azam[1], Dr. Tanvir Ahmed[2], Dr. M. Usman Hashmi[3], Rehan Ahmad[4], Abdul Manan[5], Muhammad Adrees[6], Fahad Sabah[7]
Department of CS&IT, The Superior College, Lahore, Pakistan[1, 2, 3, 5, 6, 7]
Department of Computer Science, The University of Lahore, Lahore, Pakistan[4]

***Abstract*—In this research we analyzed the performance of some well-known classification algorithms in terms of their accuracy and proposed a methodology for model stacking on the basis of their correlation which improves the accuracy of these algorithms. We selected; Support Vector Machines (svm), Naïve Bayes (nb), k-Nearest Neighbors (knn), Generalized Linear Model (glm), Latent Discriminant Analysis (lda), gbm, Recursive Partitioning and Regression Trees (rpart), rda, Neural Networks (nnet) and Conditional Inference Trees (ctree) in our research and preformed analyses on three textual datasets of different sizes; Scopus 50,000 instances, IMDB Movie Reviews having 10,000 instances, Amazon Products Reviews having 1000 instances and Yelp dataset having 1000 instances. We used R-Studio for performing experiments. Results show that the performance of all algorithms increased at Meta level. Neural Networks achieved the best results with more than 25% improvement at Meta-Level and outperformed the other evaluated methods with an accuracy of 95.66%, and altogether our model gives far better results than individual algorithms' performance.**

***Keywords*—Classification algorithms; model stacking; correlation; k-nearest neighbor; pre-processing; meta classifiers**

I. INTRODUCTION

Text classification is a method of allocating certain categories to text documents based on certain criterion. Number of classification algorithms in data mining is used to classify the appropriate class or category for text document on the basis of input algorithm used for classification. Many text classification methods are developed for efficiently solving the problem of identifying and classifying data.

The massive increase in the data being collected by information devices, needs for doing data mining and analyses on this big data, there is a need for scaling up and improving the performance of traditional data mining and learning algorithms. There exist some learning techniques with a purpose to construct a meta-classifier by joining some classifiers, usually by ensembles, voting or stacking, generated on the same data and increase the performance of algorithms [1] [2]. Grouping of the predictions of base-level classifiers with the consideration of their correlation, together with the correct class values constitute a meta-level dataset. This is the type of meta-learning which is an advanced form of stacking is addressed in this paper.

The exertion presented in this research is set in the stacking structure. Note that combining classifiers with stacking is be considered as meta-learning whereas Meta-learning means learning about learning, in practice, meta-learning takes as input results formed by learning and generalizes on them. The proposed technique can be done with tasks; (1) selection and learning of an appropriate classifier; (2) combination of predictions of base-level classifiers on the basis of correlation; (3) learning of Meta Classifiers

We proposed an extension of stacking, using an extended set of meta-level features. We show that the extension performs better than existing stacking approaches and selecting the best classifier by cross validation. The best among state-of-the-art methods is stacking with Neural Networks (nnet).

The remainder of this paper is organized as follows. Section 2 consists of literature review and surveys some other recent classification and stacking approaches and their results. Section 3 introduces our extension to stacking with correlation: the use of an extended set of meta-level features and classification via different models at the meta-level. The setup for the experiments and results of best classifiers is described in Section 4. Section 5 discusses the conclusions and future work.

II. LITERATURE REVIEW

Text widely held in a short form, which is generally used in real-time systems like news, short comment, micro-blog and numerous other fields. With the advancements in the uses of text messages, emails, online information, product reviews and movie reviews etc., data is increasing more and more. Most of the data is unusable for us while other data is important for us. So, it is required to extract the useful data from the big data. But there are number of complications with the classification of short text, for example it has irregularity, fewer features and so on.

Classification is one of the tasks most frequently carried out by so-called Intelligent Systems. Thus, a large number of techniques have been developed based on Artificial Intelligence (Logic-based techniques, Perceptron-based techniques) and Statistics (Bayesian Networks, Instance-based techniques). The goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown. This paper describes various classification algorithms and the recent attempt for improving classification accuracy-ensembles of classifiers [3].

Ensemble method is an approach to generate classifiers by applying dissimilar learning algorithms to a single dataset [4] complicated methods for combining classifiers are typically used in this setting. Model stacking is often used to learn a combining method in addition to the ensemble of classifiers [5]. To encounter the issues in classification, Jun Xiang et al. proposed a method in which they pretreated the dataset first, and then selected the important features. They used semi-supervised learning technique and Support Vector Machines (SVM) to improve the previous methods with a large number of short text datasets. They also showed a good improvement in their experimental results [6].

Prof. Purvi Rekh and Hiral Padhiyar have been attentive to the problem of short words that are used in SMS as “hpy” for “happy”, “bday” for “birthday” which decreases classification accuracy; they showed that replacement of such words with full forms, better accuracy can be achieved. They used Decision tree Algorithm for classification of SMS data as it gives better accuracy then other classifiers. But still replacing all probable short words for the given word dynamically by the full form is an issue [7].

Naïve-Bayes and k-NN classifiers are two machine learning approaches for text classification. Rocchio is the classic technique for text classification in information retrieval. Based on these three methods and using classifier combination methods, Behzad Moshiri et al. proposed a new method in text classification. This is a supervised technique in which documents are characterized as vectors and each component of the vector is connected with a particular word. They proposed voting techniques, Decision Template and OWA operator process to combine the classifiers. Their experimental results showed that the approaches decreased the error in classification to 15% whereas they used training data from 20 newsgroups dataset [8].

C.Karthika et al. proposed another text document classifier by combining the nearest neighbor classification (knn) approach with the Support Vector Machines (SVM). The objective of this study suggested SVM-NN method is to decrease the effect of parameters in classification accuracy. At training level, the SVM is applied to decrease the training samples for each of the class to their support vectors (SVs).The SVs from different classes are then used as the training data of nearest neighbor in which the distance function or similarity measures is used to calculate the which category does the testing data fits. This method also reduced time consumption [9] [24]. Another research presents a technique for enhancement explicitly intended to work with Twitter data with consideration of their structure, length and specific language; a kind of sentiment analysis. The approach used is simply extendible to other languages and capable enough to process the tweets in real time. They showed that using the training models produced with the technique described can increase the performance of sentiment classification, regardless of the domain and distribution of the test sets [10].

Another technique for improvement in accuracy of classification algorithms is ensemble method. Ensemble of classifiers, or a logical grouping of different classifiers, frequently results in improved classifications as compare to a single classifier. Though, the question about what classifiers should be selected for a given condition to create an ideal ensemble has been debated time and again. Furthermore, this technique is often computationally expensive since it requires the implementation of multiple classifiers for a single task. To provide solution of these problems, Dan Zhu et al. proposed a hybrid method for choosing and merging the models to build ensembles by incorporating Data Envelopment Analysis and stacking. Their results show the effectiveness of the proposed approach [11].

R. Mousavia et al. proposed improved Static Ensemble Selection (SES) using NSGA-II multi-objective genetic algorithm called; SES-NSGAII. The first technique in its first phase selects the best classifiers with their combiner, by immediate optimization of error and diversity objectives. In the second phase, the Dynamic Ensemble Selection-Performance (DES-P) is upgraded by using the suggested technique of first phase. The other proposed method in this research is a hybrid methodology that uses the abilities of both SES and DES methodologies and is called Improved DES-P (IDES-P). So, combining static and dynamic ensemble approaches with using NSGA-II. Results of this research approve that the proposed techniques outperform the other ensemble methods in terms of classification accuracy over 14 datasets [12].

Georgios Paliouras et al. examined the efficiency of voting and stacking. A new framework is suggested that put up famous methodologies for information extraction (IE) using stacking. To generate a meta-level data set that consists of feature vectors they performed cross-validation on the base-level data set, which contains text documents marked with related information. A classifier is then learned using the new vectors. Hence, base-level IE methods are combined with a common classifier at the meta-level. Findings of this research show that both voting and stacking are improved while using probabilistic estimates by the base-level methods. Stacking, showed consistently effective over all domains with comparably or better than voting and at all times improved than the best base-level methods [13].

Combined classification methods mutually infer all classes of a relational data set, by means of the inferences about any class label to affect inferences about related class. Kou and Cohen introduced an effective relational model on the basis of stacking that has comparable accuracy to more refined and combined inference approaches. While using experiments on both real and synthetic data, they showed that the main reason for the performance of the stacked model is the reduction in favoritism from learning the stacked model on inferred classes rather than true classes. Moreover, they revealed that the performance of the combined inference and stacked models can be recognized to an implied weighting of local and relational features at learning stage [14].

Fatemeh Nemati Koutanaeia et al. have established a three stage hybrid data mining model of feature selection and ensemble learning classification algorithms. The first stage, deals with the data collection and pre-processing. In the second stage, four Feature Selection (FS) algorithms are employed which include principal component analysis (PCA), genetic algorithm (GA), information gain ratio, and relief attribute evaluation function. Parameters setting of FS techniques is based on the accuracy resulted from the execution of the support vector machine (SVM) algorithm. Then after choosing the suitable model for every selected feature, they are applied to the base and ensemble algorithms. At this stage, the best FS algorithm with its parameters setting is specified for the next stage which is; modeling of the proposed model. At third stage, the algorithms are employed for the dataset prepared from each FS algorithm. The findings of this research showed that in the second stage, PCA is the best FS algorithm. In the third stage, the classification results indicated that the artificial neural network (ANN) adaptive boosting (AdaBoost) method has higher accuracy [15]. Some other researchers who worked for the improvement of classification algorithms used Genetic Algorithms (GAs) [16] which combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm. These algorithms have been used in machine learning and data mining applications [17], [18]. GAs has also been used in optimizing other learning techniques, such as neural networks [19].

Riyaz Sikora et al. proposed a “modified stacking ensemble machine learning algorithm using genetic algorithms”. They used data sets for their study taken from the UCI Data Repository. Five learning algorithms were used in the stacking algorithm: J48, Naïve Bayes, Neural Networks, IBk, and OneR. The best enhancement in performance was on the Chess set, where the modified stacking algorithm was able to increase the prediction accuracy by more than 10% compared to the standard stacking algorithm. The training time is also considered for both versions of the stacking algorithm. On average the modified stacking algorithm takes more time than standard stacking algorithm as it encompasses running the GA. They also proposed that training time can be significantly reduced by running the individual learning algorithms in parallel [20].

KaiquanXu et al. proposed a novel graphical model to extract and visualize comparative relations between products from customer reviews, with the interdependencies among relations taken into consideration, to help enterprises discover potential risks and further design new products and marketing strategies [22].

III. Datasets

As stated earlier we tested our proposed methodology to three pre-available datasets, IMDB Movie Reviews, Amazon Products Reviews, Yelp dataset. This section discusses these datasets in detail.

A. Scopus

The bibliographic data retrieved from the Scopus for the purpose of analysis. The data contains all types of documents published by institutes of Pakistan during 1996 to 2010. The data of each document includes author names, title, abstract, date, document type, addresses, and cited references etc. Since this study is focused on improvement in accuracy of classification algorithms and the subjected dataset is very big, we precisely extracted and analyzed the data of abstracts of publications from Scopus for some selected categories like; Computer Science, Medicine, Engineering, Agricultural & Biological Sciences and Mathematics.

B. IMDB Movie Reviews

This is a dataset for binary sentiment classification containing substantially more data than some other benchmark datasets. The core dataset contains 50,000 reviews divided evenly into 25000 train and 25000 test sets. The overall distribution of labels is balanced (25000 positive and 25000 negative). It also includes an extra 50,000 unlabeled reviews for unsupervised learning. The whole collection, does not allow more than 30 reviews for any given movie because reviews for the same movie to have associated ratings. Additionally, the training and testing sets are comprised of non-overlapping set of movies. Whole dataset has been labeled with “neg” or “pos” labels for negative and positive reviews respectively, a negative review has a score <= 4 out of 10, and a positive review has a score >= 7 out of 10. Reviews with neutral category are not included in the train/test sets. We selected 10,000 reviews (5,000 positive and 5,000 negative) in our analysis as per machine constraints.

C. Amazon Products Reviews

It comprises of sentences labeled with positive or negative sentiment, extracted from products reviews. Format: sentence \t score \n whereas the score is either 1 (for positive) or 0 (for negative). The sentences come from website: amazon.com there exist 500 positive and 500 negative sentences. Once again for this dataset, sentences that have a clearly positive or negative connotation have been selected; the goal was for no neutral sentences to be selected.

D. Yelp Dataset

This dataset contains sentences labelled with positive or negative sentiment, extracted from reviews of different restaurants Format: sentence \t score \n whereas the score is either 1 (for positive) or 0 (for negative). The sentences come from website: yelp.com there exist 500 positive and 500 negative sentences. As in earlier datasets the goal was for no neutral sentences to be selected this dataset also contains sentences that have a clearly positive or negative connotation.

IV. Tools

Getting data in structured form, preparation of data for analysis and performing analysis on data we used different tools. Tools allow various definitions, ranging from an extension of classical data mining to texts to more sophisticated formulations like “the use of large online text collections to discover new facts and trends about the world itself” [21]. Following sections discuss the tools we used during our research:

A. *Text Collector*

Text collector is a tool which integrates number of text files into single file of any format; .txt, .csv etc. By using this tool we converted the IMDB movie reviews dataset from .txt files into single .csv file.

B. *RStudio*

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. RStudio is available in open source and commercial editions and runs on the various operating systems or in a browser connected to RStudio Server or RStudio Server Pro RStudio is a tool which includes other open source software components. RStudio provides the facility to execute R code directly from the source editor. It easily manages multiple working directories using projects. RStudio has an integrated R help and documentation and interactive debugger to diagnose and fix errors quickly.

RStudio is the tool that we used for the preprocessing of data and classification of publications using different algorithms and improvement in efficiency of algorithms. RStudio includes other open source software components and libraries which includes number of predefined functions and algorithms. We used some of these functions and algorithms in our research.

V. Methodology

The following sections, discuss data set creation, feature creation from text, feature selection, base classifiers, and learning methods along with the experimental design we proposed and used for our analysis.

A. *Proposed Model*

This paper proposed a hybrid approach based on supervised learning techniques to improve the accuracy of some predictive models pre-available for text classification. Basically it is a kind of model ensembling with combining different models using stacking with consideration of separate model’s correlation and base classifier’s accuracy to allow combined predictor to get best from each model. On the basis of existing algorithms in R and correlation between these algorithms we propose the hybridization of algorithms. The algorithms were chosen on the basis of diversity of their correlation and accuracy.

As shown in Fig. 1 the subjected method is concerned with combining multiple classifiers generated by using different classification algorithms on the basis of their correlation on a single dataset S at a time. Initially a set of base-level classifiers C1, C2, . . . , CN is generated. Then, a meta-level classifier is learned using combined outputs of the base-level classifiers with actual classes and the testing dataset without class attribute.

In our proposed Hybrid Classification Algorithm; first three steps of this algorithm refers to the arrangement and pre-processing of data. The running time for these three steps depends upon the algorithm & tool used to get data structured & split data to train & test form. But on the whole, it does not affect the overall running time of this algorithm as the dominating steps for complexity of this algorithm are from 7 to 10. As far as steps 5 & 6 are concerned the running time is ϴ $(n(gi(n))$ where $gi(n)$ refer to running time of classifier i. It can be different for deferent classification techniques e.g. running time for knn, is $O\ (n)$ which is also discussed later in this section. Step 7 runs $((n-1)-1+2)$ times i.e n times while step 8 runs $n(n-(i\ +1)+2))$ times i.e. $n\ (n-i+1)$ time &step 9 runs $n(n-i+1)\ (n-1+2)$ times i.e. $n\ (n-i+1)(n+1)$ times while step 10 runs $n(n-i+1)(n+1)\ gi(n)$. So total running time after execution of first three times.

$$T(n)\ =\ n(gi(n))\ +\ n\ +\ n(n-i+1)\ +\ n(n-i+1)(n+1)\ +\ n(n-i+1)(n+1)\ gi(n) \qquad (1)$$

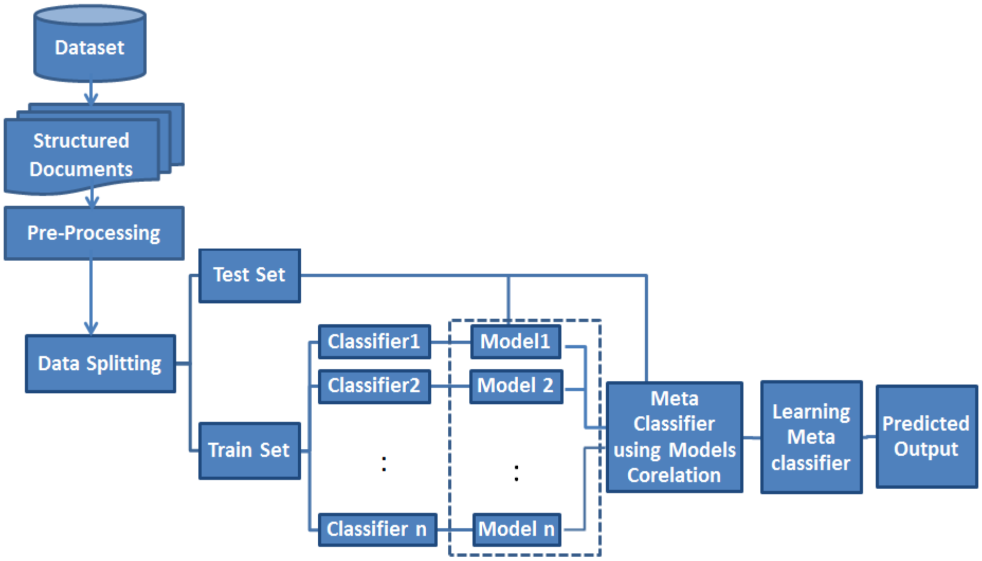

Fig. 1. Proposed Model for Text Classification.

Algorithm

```
Hybrid Classification (DataSet, v(m)[ ], fr [ ] [ ] [ ])
1. Begin
2. Structure Documents
3. Pre-processing Steps
4. Data Splitting to train & Test Sets
5. For i =1 to  n
6.       v [ i ] = cl [ i ] (Train set, test set)
7. For i =1 to n-1
8.       For j = i+1 to n
9.             For k=1 to n
10.               Fr [ i ] [ j ] [ k ] = cl [ k ] (v [ i ], v [ j ], test set,
                  actual class)
11. End
```

The simplified mathematical form of running time for steps 5 to 10 can be expressed as;

$$T(n) \ = \ O(n^3 \ gi(n)) \qquad (2)$$

Where gi (n) refers to the running time if kth classifier i.e. cl [k] this is a general form as $gi(n)$ refers to the running time of individual classifier at that particular execution time. We can be specific by taking an example of knn Classifier

KNN Algorithm

```
1. Begin
2. Input x of unknown classification
3. Set k, i < n
4. Inizinlize i = 1
5. Do Until ( k = nearest neighbours to x found)
6.       Compute distance for x to x,
7.       if ( i < k) then
8.             include $x_i$ in set of k- nearest neighbour
9.    else if ( $x_i$ classes to x than any previous nearest neighbour
) then
10.            Delete the further nearest neighbours
11.            include xi in the set of K- nearest neighbour
12.       End if
13. End Do
14. initialize i = 1
15. Do until (x assigned membership in all classes)
16.       Compute Ui (x)
17.       increment i
18. End Do
19. End
```

This algorithm shows Pseudo code of knn Algorithm from step 2 to 4, The time complexity is O (1) step 5 until step 13 has time complexity $O(n)$ time 14- has $O(1)$. Step 15 until step 18 has O (n). So that running time is

$$T(n) = O(1) + O(n) + O(1) + O(n) = O(n) \qquad (3)$$

So time complexity for knn Algorithm is O (n). When we use knn ask meta classifier the time complexity for hybrid classification algorithm become i.e.

$$T \ (n) \ = \ O \ (n^3 \ gi(n)) \qquad (4)$$

$$O \ (n^3 \ (O \ (n)) \ = \ O \ (n^4) \qquad (5)$$

As $gi(n) \ = \ O(n)$ Specifically for knn Classifier.

Working of hybrid classification algorithm first three time of algorithm refers to prepare the data as input to the classifiers used in this study i.e. get data in structured form , pre-processing steps like; cleaning, removing stop word etc. and splitting data into train and test set identifying classes. Since we are using different data set (Yelp, Amazon Reviews etc.), so these steps will be performed in all these datasets.

In step 5 and 6 different classifiers cl [i] are applied on these data sets and results is stored in vectors v [i]: as shown in Fig. 2.

These resulted vectors are then provided as input to meta classifiers along with test set and actual class in time 10. Steps 7 to 9 gives variations to classifier and to resulted vectors formed in steps 5 to 6 (Fig. 3).

On the basis of the results, calculated we can predict class for the new data more accurately as discussed in the following section.

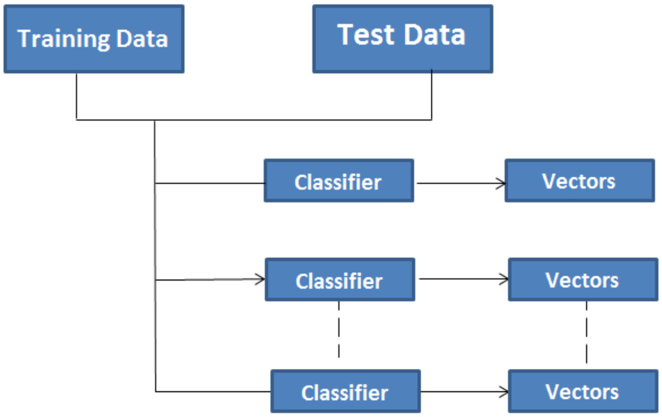

Fig. 2. Vectors Source Generation.

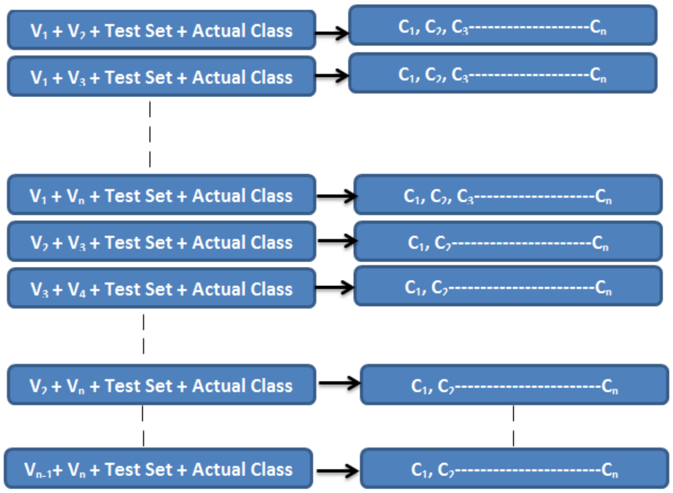
V_1 + V_2 + Test Set + Actual Class
C_1, C_2, C_3-------------------C_n
V_1 + V_3 + Test Set + Actual Class
C_1, C_2, C_3-------------------C_n
V_1 + V_n + Test Set + Actual Class
C_1, C_2, C_3-------------------C_n
V_2 + V_3 + Test Set + Actual Class
C_1, C_2----------------------C_n
V_3 + V_4 + Test Set + Actual Class
C_1, C_2----------------------C_n
V_2 + V_n + Test Set + Actual Class
C_1, C_2----------------------C_n
V_{n-1}+ V_n + Test Set + Actual Class
C_1, C_2----------------------C_n

Fig. 3. Dataset Generation for Meta Classifier.

VI. Results and Discussion

A major goal of our research was the development of an automated and effective algorithm for category detection framework that researchers, business analysts and practitioners could use to assess and infer more objective information from data obtained in large databases. In this research, we examined the classification effectiveness of both base classifiers and hybrid classifiers with in a text mining context.

The results obtained by all base-level systems in the domains of interest are initially presented in this section, Table I shows the base level classifier's accuracies for different datasets with Training and Testing 70% to 30% ratio respectively, instead of discussing in detail the individual classifiers' performance we would be investigating whether any improvement in the best results for each domain is possible at meta-level. Then, the meta-level data is analyzed, in order to determine whether and how the predictions of the base-level systems are correlated. This study is intended to serve as a basis for a comparative evaluation of voting against stacking. Then all combination methods are comparatively evaluated, while also comparing against the best base-level results. More detailed analysis of the experimental results is provided in later sections.

Table I shows that gbm, glm and lda perform better than other classifiers in case of Scopus dataset with accuracies of 67.00%, 66.33% and 63.33% respectively, in case of IMDB Movie Reviews dataset gbm, svm and glm perform better than other classifiers with accuracies of 72.92%, 72.58% and 72.33% respectively, whereas nnet, rda and svm perform better than others whereas in case of Amazon Products Reviews dataset with accuracies of 76.33%, 75.33% and 72.67% respectively and in case of Yelp dataset lda, nnet and rda give better results than other classifiers with accuracies of 69%, 68.67% and 68.33%, respectively.

We evaluated the selected methods for constructing stack of heterogeneous classifiers with stacking and shown that they perform (at best) comparably to selecting the best classifier from the stack by using their correlation values.

Table II shows the correlation between subjected algorithms for the Scopus Dataset. It can be seen that the table is symmetrical about diagonal and algorithms with negative correlations are highlighted and will be considered while stacking the algorithms. Support Vector Machines has negative correlations with k-Nearest Neighbour, Generalized Linear Model, Recursive Partitioning and Regression Trees, rda and Neural Networks, out of which Generalized Linear Model has the lowest correlation value whereas we discussed the results of stacked algorithms in next section. Naïve Bayes has negative correlations with k-Nearest Neighbour.

Table IV shows the correlation between subjected algorithms for the IMDB Movie Reviews Dataset. It can be seen that the table is symmetrical about diagonal and algorithms with negative correlations are highlighted and will be considered while stacking the algorithms. Support Vector Machines has negative correlations with k-Nearest Neighbour, Recursive Partitioning and Regression Trees, rda, Neural Networks and Conditional Inference Trees, out of which Neural Networks has the lowest correlation value whereas we discussed the results of stacked algorithms in next section. Naïve Bayes has negative correlations with k-Nearest Neighbour, gbm, rda, and Conditional Inference Trees, out of which gbm has the lowest correlation. k-Nearest Neighbour has negative correlations with Support Vector Machines, nb, Linear Discriminant Analysis and rda, out of which Linear Discriminant Analysis has the lowest correlation with k-Nearest Neighbour. Generalized Linear Model has negative correlations with rda and Neural Networks, out of which rda has the lowest correlation. Linear Discriminant Analysis has negative correlation with k-Nearest Neighbour and Conditional Inference Trees. gbm has negative correlations with nb, Recursive Partitioning and Regression Trees, rda, and Neural Networks, out of which nb has the lowest correlation. Recursive Partitioning and Regression Trees has negative correlation with Support Vector Machines, gbm and rda. rda has negative correlation with Support Vector Machines, nb, k-Nearest Neighbour, Generalized Linear Model, gbm, Recursive Partitioning and Regression Trees and Conditional Inference Trees with lowest correlation of -0.2190 with Naïve Bayes. Neural Networks has negative correlation with Support Vector Machines, Generalized Linear Model and gbm. Conditional Inference Trees has negative correlation with Support Vector Machines, nb, Linear Discriminant Analysis and rda out which Linear Discriminant Analysis has a lowest correlation.

Table V shows the results obtained from Meta Level classifiers for IMDB movie reviews dataset. In Table V each cell represents the accuracies of Meta level classifiers can be read as base classifier1 from top most row, classifier from left most column and meta classier from the lowest row. It can be seen that every algorithm at Meta level performs better than its individual performance some algorithms remarkably produces improved results as Neural Networks algorithm. Talking about the performances of these algorithms one by one; Support Vector Machines has an accuracy of 72.58% as a base classifier but when it has been stacked with different classifiers it performs better it can be seen that Support Vector Machines when stacked with Conditional Inference Trees gives 78.67% accuracy, when stacked with Neural Networks its accuracy raises to 78.75% and when stacked with Generalized Linear Model and nb it gives almost same and better results with accuracy of 78.42%. nb has an accuracy of 66.42% as a base classifier but when it has been stacked with different classifiers it performs better it can be seen that nb when stacked with Support Vector Machines gives 73.75% accuracy which is far more better than its individual accuracy, when it has been stacked with Generalized Linear Model its accuracy raises to 73.58% and when stacked with Linear Discriminant Analysis it gives results with accuracy of 73.17%. k-Nearest Neighbour has an accuracy of 61.67% as a base classifier it can be seen that when it has been stacked with different classifiers it performs better as it performs best when stacked with Conditional Inference Trees gives 73.50% accuracy, when stacked with Generalized Linear Model and Linear Discriminant Analysis its accuracy raises to 73.08% and when it has been stacked with gbm and Recursive Partitioning and Regression Trees it gives results with accuracy of 72.92%. Generalized Linear Model has an accuracy of 72.33% as a base classifier but when it has been stacked with rda gives 77.92%

accuracy, when stacked with gbm, Recursive Partitioning and Regression Trees and Neural Networks its accuracy raises to 77.83% and when it has been stacked with nb, k-Nearest Neighbour and Linear Discriminant Analysis it gives results with accuracy of 77.75%.

Linear Discriminant Analysis has an accuracy of 71.58% as a base classifier it can be seen that when it has been stacked with different classifiers it performs better as it performs best when stacked with Neural Networks gives 77.17% accuracy, when stacked with Support Vector Machines its accuracy raises to 77.08% and when it has been stacked with Generalized Linear Model it gives results with accuracy of 76.83%. gbm has an accuracy of 72.92% as a base classifier it performs best when stacked with Support Vector Machines gives 75.83% accuracy, when stacked with Linear Discriminant Analysis its accuracy raises to 75.58% and when it has been stacked with Generalized Linear Model it gives results with accuracy of 75.50%. Recursive Partitioning and Regression Trees has an accuracy of 66.25% as a base classifier it can be seen that when it has been stacked with gbm it gives 72.92% accuracy, when stacked with Support Vector Machines its accuracy raises to 72.58% and when it has been stacked with Generalized Linear Model it gives results with accuracy of 72.33%. rda has an accuracy of 70.75% as a base classifier it can be seen that when it has been stacked with Support Vector Machines it gives 76.17% accuracy, when stacked with gbm or Neural Networks its accuracy raises to 75.92% and when it has been stacked with k-Nearest Neighbour it gives results with accuracy of 75.33%. Neural Networks produces remarkably improved results with at meta level although its accuracy at base level is; 71.25 but when it has been stacked with Support Vector Machines it gives 96.58% accuracy, and when it is stacked with Generalized Linear Model it gives accuracy of 93.92% and it gives 92.33% accuracy when stacked with k-Nearest Neighbour. ctree performs best when stacked with Support Vector Machines it gives 73.08% accuracy, when it is stacked with gbm it gives 72.92% accuracy and when ctree is stacked with Generalized Linear Model or Linear Discriminant Analysis ctree gives 72.42% accuracy.

It is notable that although gbm, Generalized Linear Model, Linear Discriminant Analysis and Support Vector Machines performs better than Neural Networks at base level for the IMDB Movie Reviews dataset but the Neural Networks achieved the best results and outperformed the other evaluated methods at meta level. It achieved 96.58% accuracy when stacked with the Support Vector Machines. It is a remarkable performance considering their individual performance. From Table V, it can be seen that there is a 25.33% rise in accuracy of nnet when stacked with the svm, it also got the second highest raise of 22.67% when stacked with glm. knn stands second in rising accuracy for IMDB Movie Reviews dataset.

TABLE I. ACCURACIES OF BASE-LEVEL CLASSIFIERS FOR DIFFERENT DATASETS

	Algorithm	**Scoups Dataset**	**IMDB Movie Reviews**	**Amazon Products Reviews**	**Yelp Dataset**
1	**svm**	62.67%	72.58%	72.67%	67.33%
2	**nb**	60.67%	66.42%	50.33%	58.33%
3	**knn**	48.00%	61.67%	65.00%	60.00%
4	**glm**	66.33%	72.33%	71.00%	72.33%
5	**lda**	63.33%	71.58%	71.67%	69.00%
6	**gbm**	67.00%	72.92%	68.67%	63.67%
7	**rpart**	44.67%	66.25%	66.33%	55.00%
8	**rda**	62.67%	70.75%	75.33%	68.33%
9	**nnet**	60.83%	71.25%	76.33%	68.67%
10	**ctree**	55.00%	64.58%	67.67%	62.00%

TABLE II. CORRELATION BETWEEN SUBJECTED ALGORITHMS FOR SCOPUS DATASET

		Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Classifier 2	**svm**	X	0.0841	-0.1126	-0.3058	0.3996	0.0677	-0.2069	-0.1742	-0.0936	0.0705
	nb	0.0841	X	-0.4455	0.1130	0.0414	0.1783	0.1491	0.0176	0.2825	0.1447
	knn	-0.1126	-0.4455	X	0.0903	0.0933	-0.1845	0.0035	0.2476	-0.1955	0.0535
	glm	-0.3058	0.1130	0.0903	X	0.1479	0.1217	0.0990	0.1402	0.1048	-0.1632
	lda	0.3996	0.0414	0.0933	0.1479	X	0.0608	0.0035	0.2479	-0.0395	-0.1722
	gbm	0.0677	0.1783	-0.1845	0.1217	0.0608	X	0.1032	0.4150	-0.0100	0.0371
	rpart	-0.2069	0.1491	0.0035	0.0990	0.0035	0.1032	X	-0.0977	0.2052	0.1959
	rda	-0.1742	0.0176	0.2476	0.1402	0.2479	0.4150	-0.0977	X	-0.0585	-0.3308
	nnet	-0.0936	0.2825	-0.1955	0.1048	-0.0395	-0.0100	0.2052	-0.0585	X	0.1487
	ctree	0.0705	0.1447	0.0535	-0.1632	-0.1722	0.0371	0.1959	-0.3308	0.1487	X

TABLE III. ACCURACIES OF META-LEVEL SYSTEMS FOR SCOPUS DATASET

		Base Level Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Base Level Classifier 2	**svm**	X	67.83%	57.17%	89.67%	72.67%	77.33%	55.00%	69.83%	94.00%	63.83%
	Nb	89.33%	X	59.17%	68.17%	73.00%	71.50%	55.17%	70.00%	96.17%	64.33%
	knn	89.67%	68.17%	X	73.67%	72.50%	70.50%	44.67%	68.83%	96.17%	67.50%
	glm	89.67%	68.00%	59.67%	X	74.00%	70.67%	57.83%	70.17%	96.33%	64.83%
	Lda	89.50%	67.33%	53.67%	72.50%	X	70.50%	55.67%	69.00%	96.83%	67.00%
	gbm	90.83%	68.33%	60.00%	70.50%	73.67%	X	59.00%	71.67%	85.67%	59.83%
	rpart	87.50%	41.00%	54.00%	44.67%		70.00%	X	68.83%	95.67%	66.17%
	rda	90.17%	67.33%	55.33%	68.83%	72.83%	70.83%	54.50%	X	90.33%	61.00%
	nnet	89.33%	68.00%	55.17%	96.17%	73.83%	71.00%	55.17%	70.17%	X	61.00%
	ctree	90.67%	42.17%	54.83%	67.50%	75.33%	70.50%	52.17%	70.33%	96.83%	X
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
		Meta Level Classifier									

TABLE IV. CORRELATION BETWEEN SUBJECTED ALGORITHMS FOR IMDB MOVIE REVIEWS

		Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Classifier 2	**svm**	X	0.2346	-0.0592	0.1892	0.2502	0.0289	-0.0551	-0.0998	-0.1703	-0.1053
	nb	0.2346	X	-0.1788	0.0153	0.6190	-0.3584	0.2927	-0.2190	0.1139	-0.2964
	knn	-0.0592	-0.1788	X	0.1399	-0.3326	0.2301	0.0083	-0.2133	0.0213	0.4319
	glm	0.1892	0.0153	0.1399	X	0.2797	0.2554	0.0557	-0.1660	-0.0159	0.0868
	lda	0.2502	0.6190	-0.3326	0.2797	X	0.0024	0.1538	0.0535	0.1733	-0.4086
	gbm	0.0289	-0.3584	0.2301	0.2554	0.0024	X	-0.0890	-0.0970	-0.2891	0.3520
	rpart	-0.0551	0.2927	0.0083	0.0557	0.1538	-0.0890	X	-0.2048	0.1413	0.1564
	rda	-0.0998	-0.2190	-0.2133	-0.1660	0.0535	-0.0970	-0.2048	X	0.2474	-0.1535
	nnet	-0.1703	0.1139	0.0213	-0.0159	0.1733	-0.2891	0.1413	0.2474	X	0.2069
	ctree	-0.1053	-0.2964	0.4319	0.0868	-0.4086	0.3520	0.1564	-0.1535	0.2069	X

TABLE V. ACCURACIES OF META-LEVEL SYSTEMS FOR IMDB MOVIE REVIEWS DATASETS

		Base Level Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Base Level Classifier 2	**svm**	X	73.75%	72.50%	77.17%	77.08%	75.83%	72.58%	76.17%	96.58%	73.08%
	Nb	78.42%	X	72.67%	77.75%	76.42%	73.17%	66.42%	75.42%	85.83%	67.00%
	knn	78.17%	72.25%	X	77.75%	76.25%	73.17%	66.25%	75.33%	92.33%	64.92%
	glm	78.42%	73.58%	73.08%	X	76.83%	75.50%	72.33%	75.75%	93.92%	72.42%
	Lda	78.17%	73.17%	73.08%	77.75%	X	75.58%	71.58%	75.08%	80.75%	72.42%
	gbm	78.33%	72.92%	72.92%	77.83%	76.50%	X	72.92%	75.92%	89.92%	72.92%
	rpart	77.75%	72.25%	72.92%	77.83%	76.33%	73.08%	X	75.67%	90.92%	66.25%
	rda	77.83%	72.25%	72.67%	77.92%	76.42%	73.42%	70.75%	X	84.33%	71.67%
	nnet	78.75%	73.00%	72.58%	77.83%	77.17%	75.08%	71.25%	75.92%	X	72.17%
	ctree	78.67%	72.67%	73.50%	77.67%	76.00%	73.33%	66.25%	75.50%	84.92%	X
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
		Meta Level Classifier									

Table VI shows the correlation between subjected algorithms for the Amazon Products Reviews Dataset. It can be seen that the table is symmetrical about diagonal and algorithms with negative correlations are highlighted and will be considered while stacking the algorithms. Support Vector Machines has negative correlations with k-Nearest Neighbour, Generalized Linear Model, gbm, rda and Conditional Inference Trees, out of which rda has the lowest correlation value whereas we discussed the results of stacked algorithms in next section. Naïve Bayes has negative correlations with Generalized Linear Model, gbm, rda and Conditional Inference Trees, out of which Generalized Linear Model has the lowest correlation. k-Nearest Neighbour has negative correlations with Support Vector Machines, Generalized Linear Model, Linear Discriminant Analysis, rda and Neural Networks, out of which Neural Networks has the lowest correlation with k-Nearest Neighbour. Generalized Linear Model has negative correlations with Support Vector Machines, nb, k-Nearest Neighbour, linear Discriminant Analysis, Recursive Partitioning and Regression Trees and rda, out of which linear Discriminant Analysis has the lowest correlation. Linear Discriminant Analysis has negative correlation with k-Nearest Neighbour, Generalized Linear Model, gbm Recursive Partitioning and Regression Trees, rda and Conditional Inference Trees. gbm has negative correlations with Support Vector Machines, nb, Linear Discriminant Analysis and Neural Networks out of which nb has the lowest correlation. Recursive Partitioning and Regression Trees has negative correlation with Generalized Linear Model, Linear Discriminant Analysis, rda and Neural Networks. rda has negative correlation with Support Vector Machines, nb, k-Nearest Neighbour, Generalized Linear Model, Recursive Partitioning and Regression Trees and Neural Networks with lowest correlation of -0.2678. Neural Networks has negative correlation with k-Nearest Neighbour, gbm, Recursive Partitioning and Regression Trees, rda and Conditional Inference Trees. Conditional Inference Trees has negative correlation with Support Vector Machines, nb, Linear Discriminant Analysis and Neural Networks out which nb has a lowest correlation.

Table VII shows the results obtained from Meta Level classifiers for Amazon Products reviews dataset. Exactly same as Table III in Table VII each cell represents the accuracies of Meta level classifiers can be read as base classifier1 from top most row, classifier from left most column and meta classier from the lowest row. It can be seen that every algorithm at Meta level performs better than its individual performance some algorithms remarkably produces improved results as Support Vector Machines, Generalized Linear Model and Neural Networks algorithm.

Talking about the performances of these algorithms one by one; Support Vector Machines has an accuracy of more than 90% for all stacked models whereas it has 72.67% accuracy as a base classifier for Amazon Reviews dataset. It can be seen that Support Vector Machines when stacked with Neural Networks it gives 91.67% accuracy which is the highest one, when stacked with Conditional Inference Trees or k-Nearest Neighbour or gbm or Recursive Partitioning and Regression Trees its accuracy raises to 91.33% and when stacked with Linear Discriminant Analysis or nb it gives almost same and better results with accuracy of 91.00%. Naïve Bayes has an accuracy of 50.33% as a base classifier but when it has been stacked with different classifiers it performs better it can be seen that Naïve Bayes when stacked with Support Vector Machines or rda it gives 71.00% accuracy which is far more better than its individual accuracy, when it has been stacked with Conditional Inference Trees its accuracy raises to 70.33% and when stacked with Neural Networks it gives results with accuracy of 69.67%. k-Nearest Neighbour has an accuracy of 65.00% as a base classifier it can be seen that when it has been stacked with different classifiers it performs better as it performs best when stacked with Neural Networks gives 76.33% accuracy, when stacked with Generalized Linear Model or rda its accuracy raises to 75.33% and when it has been stacked with Support Vector Machines it gives results with accuracy of 75.00%. Generalized Linear Model has an accuracy of 71.00% as a base classifier but when it has been stacked with any of; Naïve Bayes, k-Nearest Neighbour, gbm, Recursive Partitioning and Regression Trees, Neural Networks or Conditional Inference Trees it gives 91.67% accuracy, when stacked with Linear Discriminant Analysis its accuracy raises to 91.00% and when it has been stacked with Support Vector Machines it gives results with accuracy of 90.00%.

Linear Discriminant Analysis has an accuracy of 71.67% as a base classifier it can be seen that when it has been stacked with different classifiers it performs better as it performs best when stacked with Neural Networks gives 88.00% accuracy, when stacked with rda its accuracy raises to 87.67% and when it has been stacked with gbm it gives results with accuracy of 87.00%. gbm has an accuracy of 68.67% as a base classifier it performs best when stacked with Neural Networks gives 78.00% accuracy, when stacked with rda its accuracy raises to 76.67% and when it has been stacked with Linear Discriminant Analysis or Support Vector Machines it gives same results with accuracy of 75.67%. Recursive Partitioning and Regression Trees has an accuracy of 66.33% as a base classifier it can be seen that when it has been stacked with Neural Networks it gives 76.33% accuracy, when stacked with rda its accuracy raises to 75.33% and when it has been stacked with Support Vector Machines it gives results with accuracy of 72.67%.rda has an accuracy of 75.33% as a base classifier it can be seen that when it has been stacked with Neural Networks it gives 83.00% accuracy, when stacked with Support Vector Machines or Linear Discriminant Analysis or gbm its accuracy raises to 76.67% and when it has been stacked with Generalized Linear Model or Recursive Partitioning and Regression Trees or Conditional Inference Trees it gives results with accuracy of 76.33%.

Neural Networks produces remarkably improved results with at meta level although its accuracy at base level is; 76.33 but when it has been stacked with all classifiers except; Linear Discriminant Analysis and gbm it gives 92.33% accuracy, and when it is stacked with Linear Discriminant Analysis or gbmit gives accuracy of 92.00%. Conditional Inference Trees performs best when stacked with rda or Neural Networks it gives 76.33% accuracy, when it is stacked with Recursive Partitioning and Regression Trees it gives 75.33% accuracy and when ctree is stacked with Support Vector Machines it gives 72.67% accuracy. Although its accuracy as individual classifier is 67.67%.

Neural Networks at base level for the Amazon Products Reviews dataset has the highest accuracy and it has achieved the best results and outperformed the other evaluated methods at meta level. But other base classifiers like; Support Vector Machines, Generalized Linear Model and Linear Discriminant Analysis also gives remarkable results as compared to their individual performances.

From Table VII it can be seen that there is a glm and Naïve Bayes got highest raise in accuracy which is 20.67%. svm also improved a lot and got raise of 19% as its highest improvement. Although nnet at Meta level for Amazon Products Reviews once again outperforms all other classifiers but it had not got such improvement as glm, Naïve Bayes and svm acquired.

Table VIII shows the correlation between subjected algorithms for the Yelp Dataset. It can be seen that the table is symmetrical about diagonal and algorithms with negative correlations are highlighted and will be considered while stacking the algorithms. Support Vector Machines has negative correlations with Generalized Linear Model, Linear Discriminant Analysis, gbm, Recursive Partitioning and Regression Trees, rda, Neural Networks and Conditional Inference Trees, out of which gbm has the lowest correlation value whereas we discussed the results of stacked algorithms in next section. Naïve Bayes has negative correlations with gbm, rda, Neural Networks and Conditional Inference Trees, out of which rda has the lowest correlation. k-Nearest Neighbour has negative correlations with gbm, Recursive Partitioning and Regression Trees, and Conditional Inference Trees, out of which Conditional Inference Trees has the lowest correlation with k-Nearest Neighbour. Generalized Linear Model has negative correlations with Support Vector Machines, gbm, Recursive Partitioning and Regression Trees, rda, Neural Networks and Conditional Inference Trees, out of which Support Vector Machines has the lowest correlation. Linear Discriminant Analysis has negative correlation with Support Vector Machines, Neural Networks and Conditional Inference Trees. gbm has negative correlations with Support Vector Machines, nb, k-Nearest Neighbour, Generalized Linear Model, and rda, out of which Support Vector Machines has the lowest correlation. Recursive Partitioning and Regression Trees have negative correlation with Support Vector Machines, k-Nearest Neighbour, Generalized Linear Model, rda, Neural Networks and Conditional Inference Trees. rda has negative correlation with Support Vector Machines, nb, Generalized Linear Model, gbm, and Recursive Partitioning and Regression Trees with lowest correlation of -0.2253 with Naïve Bayes. Neural Networks has negative correlation with Support Vector Machines, nb, Linear Discriminant Analysis and Recursive Partitioning and Regression Trees. Conditional Inference Trees has negative correlation with Support Vector Machines, nb, k-Nearest Neighbour, Generalized Linear Model, Linear Discriminant Analysis and Recursive Partitioning and Regression Trees out which Linear Discriminant Analysis has a lowest correlation.

Table IX shows the results obtained from Meta Level classifiers for Yelp dataset. Similarly as Table III, Table V and Table VII in Table IX each cell represents the accuracies of Meta level classifiers can be read as base classifier1 from top most row, classifier from left most column and meta classier from the lowest row. It can be seen that every algorithm at Meta level performs better than its individual performance some algorithms remarkably produces improved results as Neural Networks algorithm.

Talking about the performances of these algorithms one by one; Support Vector Machines has an accuracy of 67.33% as a base classifier but when it has been stacked with different classifiers it performs better it can be seen that Support Vector Machines when stacked with Generalized Linear Model gives 88.67% accuracy, when stacked with Naïve Bayes or gbm or Neural Networks its accuracy raises to 88.33% and when stacked with Recursive Partitioning and Regression Trees and Conditional Inference Trees it gives almost same and better results with accuracy of 88.00%. Naïve Bayes has an accuracy of 58.33% as a base classifier but when it has been stacked with different classifiers it performs better it can be seen that nb when stacked with Generalized Linear Model gives 62.33% accuracy which is far more better than its individual accuracy, when it has been stacked with Support Vector Machines or rda its accuracy raises to 62.00% and when stacked with Linear Discriminant Analysis or Neural Networks it gives results with accuracy of 61.33%.

k-Nearest Neighbour has an accuracy of 60% as a base classifier it can be seen that when it has been stacked with different classifiers it performs better as it performs best when stacked with Generalized Linear Model gives 75.33% accuracy, when stacked with Linear Discriminant Analysis its accuracy raises to 71.67% and when it has been stacked with Support Vector Machines or Neural Networks it gives results with accuracy of 71.00%. Generalized Linear Model has an accuracy of 72.33% as a base classifier but when it has been stacked with Naïve Bayes gives 89.33% accuracy, when stacked with gbm, k-Nearest Neighbour or Neural Networks its accuracy raises to 88.67% and when it has been stacked with Support Vector Machines, Recursive Partitioning and Regression Trees, rda or Conditional Inference Trees it gives results with accuracy of 88.33%.

Linear Discriminant Analysis has an accuracy of 69.00% as a base classifier it can be seen that when it has been stacked with different classifiers it performs better as it performs best when stacked with Support Vector Machines or Naïve Bayes or ctree gives 85.33% accuracy, when stacked with k-Nearest Neighbour or Generalized Linear Model or Recursive Partitioning or Regression Trees or rda its accuracy raises to 85.00% and when it has been stacked with gbm it gives results with accuracy of 83.67%. gbm has an accuracy of 63.67% as a base classifier it performs best when stacked with Generalized Linear Model gives 74.67% accuracy, when stacked with Neural Networks its accuracy raises to 72.00% and when it has been stacked with Linear Discriminant Analysis it gives results with accuracy of 69.33%. Recursive Partitioning and Regression Trees has an accuracy of 55.00% as a base classifier it can be seen that when it has been stacked with Linear Discriminant Analysis it gives 69.00% accuracy, when stacked with Neural Networks its accuracy raises to 68.67% and when it has been stacked with rda it gives results with accuracy of 68.33%. rda has an accuracy of 68.33% as a base classifier it can be seen that when it has been stacked with Generalized Linear Model or Neural Networks it gives 73.67% accuracy, when stacked with Linear Discriminant Analysis its accuracy raises to 72.33% and when it has been stacked with Support Vector Machines or Recursive Partitioning and Regression Trees or ctree it gives results with accuracy of 69.00%.

TABLE VI. CORRELATION BETWEEN SUBJECTED ALGORITHMS FOR AMAZON PRODUCT REVIEWS

		Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Classifier 2	**svm**	1.0000	0.2497	-0.1361	-0.0737	0.1340	-0.0168	0.0430	-0.2395	0.2976	-0.2189
	nb	0.2497	1.0000	0.1007	-0.5704	0.0471	-0.2166	0.0795	-0.2977	0.1876	-0.3439
	knn	-0.1361	0.1007	1.0000	-0.0160	-0.1508	0.0680	0.2929	-0.1714	-0.4354	0.1879
	glm	-0.0737	-0.5704	-0.0160	1.0000	-0.3277	0.1365	-0.0824	-0.0158	0.0186	0.1538
	lda	0.1340	0.0471	-0.1508	-0.3277	1.0000	-0.0018	-0.1847	-0.0317	0.3914	-0.2228
	gbm	-0.0168	-0.2166	0.0680	0.1365	-0.0018	1.0000	0.1992	0.0684	-0.1427	0.5784
	rpart	0.0430	0.0795	0.2929	-0.0824	-0.1847	0.1992	1.0000	-0.0177	-0.1555	0.2982
	rda	-0.2395	-0.2977	-0.1714	-0.0158	-0.0317	0.0684	-0.0177	1.0000	-0.2678	0.0376
	nnet	0.2976	0.1876	-0.4354	0.0186	0.3914	-0.1427	-0.1555	-0.2678	1.0000	-0.1654
	ctree	-0.2189	-0.3439	0.1879	0.1538	-0.2228	0.5784	0.2982	0.0376	-0.1654	1.0000

TABLE VII. ACCURACIES OF META-LEVEL SYSTEMS FOR AMAZON PRODUCT REVIEWS DATASETS

		Base Level Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Base Level Classifier 2	**svm**	X	71.00%	75.00%	90.00%	86.67%	75.67%	72.67%	76.67%	92.33%	72.67%
	nb	91.00%	X	72.00%	91.67%	85.67%	72.33%	66.33%	75.33%	92.33%	67.67%
	knn	91.33%	64.67%	X	91.67%	85.67%	73.67%	67.00%	75.33%	92.33%	72.33%
	glm	91.00%	68.67%	75.33%	X	85.33%	73.67%	71.00%	76.33%	92.33%	71.67%
	lda	91.00%	69.00%	72.00%	91.00%	X	75.67%	71.67%	79.67%	92.00%	68.67%
	gbm	91.33%	67.67%	72.00%	91.67%	87.00%	X	68.67%	76.67%	92.00%	67.67%
	rpart	91.33%	68.33%	73.00%	91.67%	85.67%	71.00%	X	76.33%	92.33%	75.33%
	rda	90.33%	71.00%	75.33%	86.33%	87.67%	76.67%	75.33%	X	92.33%	76.33%
	nnet	91.67%	69.67%	76.33%	91.67%	88.00%	78.00%	76.33%	83.00%	X	76.33%
	ctree	91.33%	70.33%	72.67%	91.67%	85.67%	69.67%	69.67%	76.33%	92.33%	X
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
		Meta Level Classifier									

TABLE VIII. CORRELATION BETWEEN SUBJECTED ALGORITHMS FOR YELP DATASET

	svm	**nb**	**Knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
svm	1.0000	0.2683	0.2940	-0.2098	-0.0528	-0.4631	-0.2577	-0.0581	-0.0776	-0.0312
nb	0.2683	1.0000	0.1602	0.1623	0.1807	-0.0290	0.0897	-0.2253	-0.1466	-0.0169
knn	0.2940	0.1602	1.0000	0.3066	0.0170	-0.1122	-0.1745	0.2178	0.0227	-0.1786
glm	-0.2098	0.1623	0.3066	1.0000	0.1036	-0.1541	-0.0655	-0.0227	0.0757	-0.1758
lda	-0.0528	0.1807	0.0170	0.1036	1.0000	0.0478	0.3140	0.0922	-0.2034	-0.3327
gbm	-0.4631	-0.0290	-0.1122	-0.1541	0.0478	1.0000	0.3848	-0.0032	0.0539	0.2208
rpart	-0.2577	0.0897	-0.1745	-0.0655	0.3140	0.3848	1.0000	-0.1085	-0.1425	-0.0548
rda	-0.0581	-0.2253	0.2178	-0.0227	0.0922	-0.0032	-0.1085	1.0000	0.0848	0.0722
nnet	-0.0776	-0.1466	0.0227	0.0757	-0.2034	0.0539	-0.1425	0.0848	1.0000	0.1223
ctree	-0.0312	-0.0169	-0.1786	-0.1758	-0.3327	0.2208	-0.0548	0.0722	0.1223	1.0000

TABLE IX. ACCURACIES OF META-LEVEL SYSTEMS FOR YELP DATASETS

		Base Level Classifier 1									
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
Base Level Classifier 2	**svm**	X	62.00%	71.00%	88.33%	85.33%	67.67%	67.33%	69.00%	94.33%	67.33%
	nb	88.33%	X	60.33%	89.33%	85.33%	64.00%	62.67%	68.67%	94.33%	62.00%
	knn	87.33%	60.67%	X	88.67%	85.00%	64.67%	60.00%	68.67%	91.00%	72.33%
	glm	88.67%	62.33%	75.33%	X	85.00%	74.67%	72.33%	73.67%	94.33%	69.00%
	lda	87.67%	61.33%	71.67%	88.00%	X	69.33%	69.00%	72.33%	94.33%	65.00%
	gbm	88.33%	60.33%	61.67%	88.67%	83.67%	X	63.67%	68.33%	94.33%	62.00%
	rpart	88.00%	58.33%	60.67%	88.33%	85.00%	63.67%	X	69.00%	93.33%	68.33%
	rda	87.00%	62.00%	62.67%	88.33%	85.00%	68.33%	68.33%	X	94.33%	68.67%
	nnet	88.33%	61.33%	71.00%	88.67%	83.33%	72.00%	68.67%	73.67%	X	70.67%
	ctree	88.00%	59.33%	61.33%	88.33%	85.33%	65.00%	62.00%	69.00%	94.00%	X
		svm	**nb**	**knn**	**glm**	**lda**	**gbm**	**rpart**	**rda**	**nnet**	**ctree**
		Meta Level Classifier									

Neural Networks produces remarkably improved results with at meta level although its accuracy at base level is; 71.25 but when it has been stacked with all except k-Nearest Neighbour and ctree it gives 94.33% accuracy, and when it is stacked with ctree it gives accuracy of 94.00% and it gives 91.00% accuracy when stacked with k-Nearest Neighbour. ctree performs best when stacked with Neural Networks it gives 70.67% accuracy, when it is stacked with Generalized Linear Model it gives 69.00% accuracy and when ctree is stacked with rda the ctree gives 68.67% accuracy. Once again Neural Networks achieved the best results and outperformed the other evaluated methods at meta level. It achieved 94.33% accuracy when stacked which is a remarkable performance considering its individual performance.

From Table IX, it can be seen that there is a 25.66% rise in accuracy of nnet when stacked with the svm, nb, glm, lda, gbm and rda it also got the second highest raise of 25.33% when stacked with ctree. svm stands second in rising accuracy for Yelp dataset.

VII. CONCLUSIONS

In this research we presented a modified version of the standard stacking algorithm that uses a correlation between algorithms to create a Meta classifier. We tested the individual learning algorithms and Meta classifiers over different textual datasets; IMDB Movie Reviews, Amazon Product Reviews and Yelp Dataset and showed the improvement in performance over the individual learning algorithms as well as over the standard stacking algorithm. We have concluded that our approach performs better than other mentioned document classification approaches with a highest improvement of 25.66% in Yelp Dataset and 96.58% accuracy for IMDB Movie Reviews. The proposed solution can be of good use in many intelligence applications.

REFRENCES

[1] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. "Journal of artificial intelligence research", 11, 169-198.

[2] Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one?. "Machine learning", 54(3), 255-273.

[3] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. "Artificial Intelligence Review", 26(3), 159-190.

[4] Merz, C. J. (1999). Using correspondence analysis to combine classifiers. "Machine Learning", 36(1-2), 33-58.

[5] Wolpert, D. H. (1992). Stacked generalization. "Neural networks", 5(2), 241-259.

[6] Yin, C., Xiang, J., Zhang, H., Yin, Z., & Wang, J. (2015). Short text classification algorithm based on semi-supervised learning and SVM. "International Journal of Multimedia and Ubiquitous Engineering", 10(12), 195-206.

[7] Padhiyar, H., & Padhiar, D. N. (2013). Improving Accuracy of Text Classification for SMS Data. International Journal for Scientific Research & Development, 1(10), 181-189.

[8] Danesh, A., Moshiri, B., & Fatemi, O. (2007, July). Improve text classification accuracy based on classifier fusion methods. Information Fusion, 2007 10th International Conference on (pp. 1-6). IEEE.

[9] Sivakumar, M., Karthika, C., & Renuga, P. (2014). A Hybrid Text Classification Approach using KNN and SVM. Int. J. Innov. Res. Sci. Eng. Technol, 3(3), 1987-1991.

[10] Balahur, A. (2013). Sentiment analysis in social media texts. In Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 120-128).

[11] Mousavi, R., & Eftekhari, M. (2015). A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches. Applied Soft Computing, 37, 652-666.

[12] Sigletos, G., Paliouras, G., Spyropoulos, C. D., & Hatzopoulos, M. (2005). Combining information extraction systems using voting and stacked generalization. Journal of Machine Learning Research, 6(Nov), 1751-1782.

[13] Fast, A., & Jensen, D. (2008, December). Why stacked models perform effective collective classification. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 785-790). IEEE.

[14] Koutanaei, F. N., Sajedi, H., & Khanbabaei, M. (2015). A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. Journal of Retailing and Consumer Services, 27, 11-23.

[15] Goldberg, D. E. (1989). Genetic Alogorithms in Search. Optimization & Machine Learning.

[16] Agustı, L. E., Salcedo-Sanz, S., Jiménez-Fernández, S., Carro-Calvo, L., Del Ser, J., & Portilla-Figueras, J. A. (2012). A new grouping genetic algorithm for clustering problems. Expert Systems with Applications, 39(10), 9695-9703.

[17] Sikora, R., & Piramuthu, S. (2005). Efficient genetic algorithm based data mining using feature selection with Hausdorff distance. Information Technology and Management, 6(4), 315-331.

[18] Sexton, R. S., Sriram, R. S., & Etheridge, H. (2003). Improving decision effectiveness of artificial neural networks: a modified genetic algorithm approach. Decision Sciences, 34(3), 421-442.

[19] Sikora, R. (2015). A modified stacking ensemble machine learning algorithm using genetic algorithms. In Handbook of Research on Organizational Transformations through Big Data Analytics (pp. 43-53). IGi Global.

[20] Hearst, M. A. (1999, June). Untangling text data mining. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (pp. 3-10). "Association for Computational Linguistics".

[21] Xu, K., Liao, S. S., Li, J., & Song, Y. (2011). Mining comparative opinions from customer reviews for Competitive Intelligence. "Decision support systems", 50(4), 743-754.

[22] Ting, K. M., & Witten, I. H. (1999). Issues in stacked generalization. Journal of artificial intelligence research, 10, 271-289.

[23] Karman, S. S., & Ramaraj, N. (2008). Similarity-Based Techniques for Text Document Classification. Int. J. SoftComput, 3(1), 58-62.