Genetic Algorithm and Ensemble Learning Aided Text Classification using Support Vector Machines

Anshumaan Chauhan, Ayushi Agarwal, Razia Sulthana
Department of Computer Science and Engineering
Birla Institute of Technology and Science Pilani
Dubai, UAE

***Abstract*—Text classification is one of the areas where machine learning algorithms are used. The size of the dataset and the methods used for converting the textual words into vectors play a major role in classifying them. This paper proposes a heuristic based approach to classify the documents using Genetic Algorithm aided Support Vector Machines (SVM) and Ensemble Learning approach. The real valued representation of the textual data into vectors is done on applying Term Frequency – Inverse Document Frequency (TF-IDF) and Bi-Normal Separation (BNS). However, in this paper, the common data misclassification issue in SVM is overcome by introducing two algorithms that adds weightage to accurate classification. The first algorithm applied BNS and TF-IDF along with ensemble learning and constructs a voting classifier for classifying the textual documents. The results produced justify that TF-IDF produces good results with voting classifier than BNS for classification. Henceforth TF-IDF is applied in the subsequent approach for vector generation. Secondly, genetic algorithm is applied along with OneVsRest strategy in SVM to overcome the drawback of multiclass multilabel classification. The results show that Genetic algorithm improves the accuracy of classification even with a very small labelled dataset, as genetic algorithm applies the process of Mutation and Cross over across many generations to understand the pattern of right classification.**

Keywords*—Genetic algorithm; ensemble learning; support vector machines; text classification

I. Introduction

Text classification is a primary domain focused by companies that handle big data and big data related applications. To search the relevant and related documents, these companies evolve new methods to classify and organize their data. Extensive collection of research articles, newspapers, journals and other related textual content are available online to be classified in various categories. It still remains a cumbersome process to sort them with prevailing algorithms as they are time consuming and are inefficient. However, machine learning algorithms have a forte in these kinds of applications. There exist many supervised algorithms such as Naïve Bayes (Probabilistic Generative classifier), Decision Trees (DT), Random Forest (RF), SVM which are used for text classification. Though in earlier days SVM were not used for text classification, in earlier 2000s, it contended with neural networks producing the most desirable results [1]. SVM is compared with Logistic Regression [2], Naïve Bayes and K-Nearest Neighbor (KNN) [3][4] where it returned a satisfactory recall, F1-score and accuracy. SVM outperforms Gaussian kernel and Naïve Bayes by producing a root mean squared error of 15.7% and 22.62% respectively [5]. Another article [6] compares SVM with NB and proves that the former provides better results than latter.

Text classification is a sub-area of Natural Language Processing, where a classifier is used to study a text and then assign a category to that document. It is also known as text categorization or document categorization. There is a lot of data out there which is not eligible for use until it is classified into a proper category and gets structured and organized. Data is properly categorized after doing text classification can be used in many places such as Sentimental analysis, Movie Reviews and many more.

The main contributions in this work are:

1) BNS and TF-IDF along with Ensemble Learning is applied to reduce the misclassification and increase the accuracy of the SVM model. This model is tested on a Spam text classification dataset from Kaggle which is a binary dataset. The results give the inference that TF-IDF will always work better than BNS scaling and the proposed model where TF-IDF along with the Voting Classifier is used shows the best performance.

2) Genetic Algorithm along with OneVsRest classifier is applied to increase the performance of labelled dataset when the provided dataset has limited labelled tuples and more unlabelled tuples. This enhances the performance of classifying multilabel multi-instance classification. It is tested on the Reuters dataset.

The organization of the rest of the paper is given here: Section 2 contains Literature survey of the research regarding the subject. Section 3 comprises the methodology of the proposed approaches. Section 4 contains the evaluated results. Section 5 concludes the paper.

II. Literature Survey

Text classification is the process of categorizing documents. This task is imperative in companies that demand classifying data to ease managing them. SVM is a contemporary approach that is applied to classify the textual content. Although, SVM outperforms a number of algorithms, it becomes challenging to draw the decision boundary as SVM requires identifying the support vectors and then classifying the data. Articles [7][8][9] applies feature scaling using TF-

IDF and neglects to scale the words appropriately. The article [7] came up with a new method of converting words into vectors known as BNS. BNS scaling is a weighting term proposed by HP labs and is applied in most of the research articles. In it BNS score for each of the feature words is calculated and then TF-BNS is used instead of IDF for a better scaling. The difference value between the inverse normal cumulative function of TPR (true positive rate) and FPR (false positive rate) is used as a BNS score. When tested, performance metrics of BNS were better than all other algorithms such as TF, IDF and TF*IDF. It's also a proven fact that, TF and IDF when applied individually delivered better accuracy than TF*IDF.

The article [8] applies the weighting algorithm TF-RF (Term frequency relevance frequency) which proved to be better than TF-IDF as it improves the effect of identifying the discriminating terms. Approaches like Word2vec (a google product) or latent semantic indexing is combined with TF-IDF to bring an extra feature that helps to train SVM further for text classification [10]. However, it becomes arduous for SVM to classify the data, if there are not enough features or the dataset is not big enough for training [11][12].

Hybrid models and hierarchical models were developed to overcome the problem of misclassification of data points that lie near the decision boundary. SVM is combined with decision trees in [13][14], with random forest in [15] and both the models were too complex and took increased training time. The former applied SVM in each node of the Decision tree to separate some group of/individual classes from the rest. Accuracy score of DT+SVM was better than Naïve Bayes, standard SVM and standard DT. The latter extended this idea and applied RF. This model reduced the misclassifications near the decision boundary and performed exceptionally well for only large datasets. However, when this method was tested on an average sized dataset, it did not have a good accuracy score. SVM in hierarchies were applied in [16][17] and produced trivial results than those produced by decision trees and random forest.

A hybrid model of SVM with KNN (K-Nearest Neighbor) is proposed in [18]. The concept of incremental learning is applied to speed up the process of training. Results showed that the F1-score of the hybrid model was better than standard SVM as well as standard KNN.

It's well known that SVM is a supervised ML algorithm which can show remarkable performance when executed with appropriate kernels. An experiment using 4 different kernels-Linear, Polynomial, Gaussian and Sigmoid is done in [19]. Results showed that the Gaussian (Radial Basis Function) kernel produced good results for text classification. For multiclass classification, either OneVsOne and OneVsRest classifiers are preferably applied. The author in [19] substantiates that OneVsRest is much more robust when it comes to classification of categories that have a very small number of documents for training. OneVsRest was compared with OneVsOne for multiclass classification in [20] and the author compared four different algorithms for multiclass classification and accuracy was measured. OneVsRest showed 98% accuracy and proved better than other algorithms.

III. Methodology

In the proposed model, SVM, BNS, AdaBoost and Voting Classifier are applied for text classification. A brief description of all the algorithms used is given here.

A. Support Vector Machine

SVM, a supervised ML algorithm is applied both in Regression and classification. SVM creates a hyperplane/decision boundary for a 2D/nD data, such that classes are separated as widely as possible. It identifies the support vectors to create a margin that is as wide as possible (Fig. 1).

Compared to other classification algorithms SVM is much faster, accurate and handles well the problem of overfitting. SVM being a binary classifier limits it for certain applications. Yet SVM can be combined with many other algorithms such as OneVsOne and OneVsRest which enables it for multiclass classification. In the proposed algorithm, OneVsRest algorithm is applied over Reuters dataset.

B. Bi-Normal Separation

BNS is applied for term weighting and overcomes the problem faced by IDF (Inverse Document Frequency), i.e., inappropriate scaling of some terms. Formula used by them for assigning a score to a word is given here.

$$|f(TPR)-f(FPR)| \quad (1)$$

f is the inverse normal cumulative distribution function.

Let p = total documents categorized as positive class in training dataset, n = total documents categorized as negative class in training dataset, tp = True positives: number of documents of positive class label which comprises the 'word', fp = False positives: number of documents of negative class label which comprises the 'word'.

$$fn=p-tp,\ tn=n-fp \quad (2)$$

tpr = Probability that word is present in a document, given the document belongs to the positive class label.

$$tpr=tpp \quad (3)$$

fpr = Probability that word is present in a document, given the document belongs to the negative class label

$$fpr=fpn \quad (4)$$

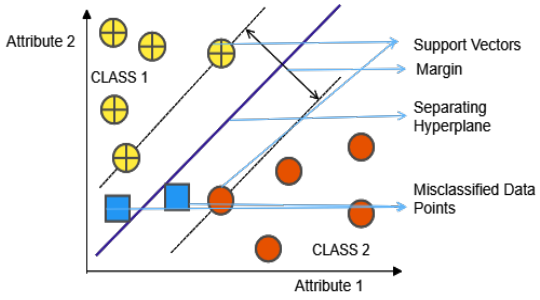

Fig. 1. Support Vector Machines.

C. AdaBoost

Ensemble Learning is a type of learning which combines many weak classifiers to form a strong classifier. Weak classifiers have low accuracy and strong classifiers have high accuracy. Ensemble Learning is of 3 different ways, i.e., Bagging, Boosting and Stacking. In the proposed method Boosting is applied. Boosting is an Ensemble Learning approach which helps us to create more accurate classifiers with minimal error. Boosting algorithms are less prone to overfitting problems. AdaBoost is a boosting algorithm which aims to fit the training set in a better and more accurate way after each iteration/pass (to other classifiers). It constantly increases the weights of misclassified points, and this modified dataset with modified weights is passed onto the next weak classifier. This is a sequential kind of process, where the output of one classifier becomes the input of another (Fig. 2). This process will terminate when the maximum number of classifiers are reached, or the dataset fits completely.

D. Voting Classifier

Voting Classifier works similar to real life elections. A person who gets the majority of votes wins and in a similar way the outcome which is predicted by majority of classifiers is given as final output. It strives to increase accuracy. Since predictions of many classifiers are being considered, voting classifiers help in decreasing the error rate thereby minimizing the chance of misclassification. In (Fig. 3), a simple working of Voting Classifier is explained.

E. Classification using Ensemble Learning on applying TF-IDF and BNS

One of the prevalent challenges of SVM mentioned by many research papers is that SVM does misclassification when the test data tuple lies within the area of hyperplane. To overcome this drawback, a Voting classifier along with the AdaBoost algorithm is used so that the misclassifications would be reduced.

1) Dataset: The spam text classification dataset was chosen to test the proposed method 1. The dataset is split into 2 halves. 80% of the dataset is used for training purposes and 20% of the dataset is used for testing purposes. The Table I shows the train test split ratio of the number of tuples.

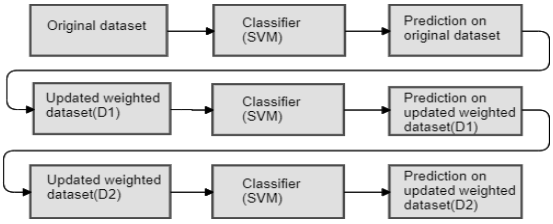

Fig. 2. Adaboost with SVM.

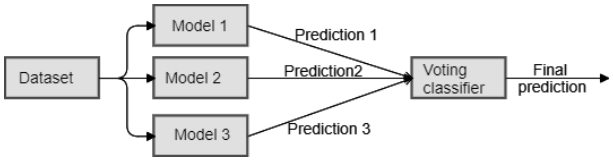

Fig. 3. Voting Classifier.

TABLE I. TRAINING-TEST DATASET SPLIT

Dataset splits	*Number of tuples*
X_{train}	4457
Y_{train}	4457
X_{test}	1115
Y_{test}	1115

2) Proposed ensemble algorithm: As already mentioned about the key features of BNS, BNS scaling for feature scaling/term weighting (word to vector) is applied to improve the accuracy of SVM. Following steps are done in the proposed algorithm:

Step 1: Firstly, preprocessing (Fig. 4) is applied on the whole initial dataset D.

Step 2: Now the preprocessed dataset D is split into 2 datasets: D_{train} and D_{test}. Following which, the featured words are converted into vectors using both TF-IDF and BNS, in order to compare the difference in accuracy.

Step 3: AdaBoost classifier that uses SVM with kernel as RBF function (Gaussian function) is applied. The classifier takes a number of iterations to fit the dataset.

Step 4: Succeeding, a Random Forest is created. Though random forest takes a longer time to train and predict as compared to a single decision tree classifier, it also has the advantage that it gives better accuracy.

Step 5: In the final ensemble model: AdaBoost, Random Forest, SVM and Decision Tree are combined together. This ensemble model is used as Voting classifier. The generated ensemble model is trained and the results of the prediction are obtained.

Step 6: Standard SVM with RBF kernel is applied on both the datasets (dataset converted using BNS and dataset converted using TF-IDF) and the results of the prediction are obtained.

Step 7: The accuracy score of both the classifiers 'Voting Classifier' and 'SVM with RBF kernel' is compared.

In the (Fig. 5) above, you can get a better diagrammatic representation of framework-1.

In this Section, ensemble learning along with TF-IDF as well as BNS is applied and the results of TF-IDF were better than BNS. Henceforth in the next section TF-IDF is applied to classify multi class dataset using SVM.

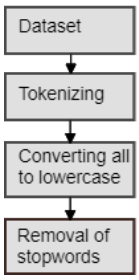

Fig. 4. Steps Involved in Pre-processing.

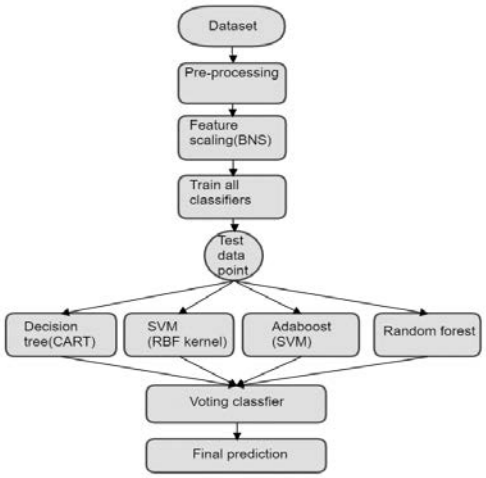

Fig. 5. Framework-1.

F. Genetic Algorithm over SVM

Genetic Algorithm (GA) has been used in many Artificial Intelligence (AI) applications for generating dataset. They were used in many applications such as training of Recurrent Neural Networks, GBML (Genetic Based Machine Learning Algorithm) and DNA Analysis, etc.

This concept is applied in the framework to create a sample labelled dataset which can be used to train the Machine Learning algorithm in a much better manner. There are many instances when due to unavailability of proper labelled dataset, the accuracy is not to satisfiability level. By unavailability, it means reduced number of tuples for training a particular class label or reduced features and comparatively smaller number of tuples. The framework aims to solve the hypothesis where the tuples are more, with few labelled and rest un-labelled.

1) Dataset: To create a temporary label for some random un-labelled tuples, which can further help us to achieve greater stability, the following steps are applied over the Reuters dataset. It has a total of 7769 documents for training purposes and 3019 documents for testing purposes. Out of these 7769, 2590 documents are taken as labelled and out of the remaining 5179 documents, 740 documents are taken as unlabeled ones. For these unlabeled ones we will run the genetic algorithm and try to create a label which will be as close to the real label as possible. Reuters dataset has a total of 90 classes in it. The dataset split is shown in Fig. 6 and Fig. 7.

On applying K- Nearest Neighbors and Decision Trees over 3019 documents the accuracy obtained was 8.07 and 9.13 respectively. This poor accuracy value motivated us to apply genetic algorithm over SVM for further classifying the tuples. Our framework improves the accuracy of classification.

2) Proposed genetic algorithm for classification: Objective: To label the unlabeled tuples and to arrive in generating an SVM model that classifies the tuples with maximal accuracy. It is implemented over Reuters Dataset.

Step 1: Split the initial dataset D into two parts: $D_{labeled}$ and $D_{unlabeled}$. As Reuters is a labelled dataset, tuples (text document) are randomly picked from the Reuters dataset, remove their labels and store in $D_{unlabeled}$.

Step2: To convert the labels of the dataset $D_{labeled}$ to numeric values, MultiLabelBinarizer is used (As the objective is to perform Multi class classification).

Step 3: Text documents of datasets $D_{labeled}$ and $D_{unlabeled}$, are preprocessed, i.e., operations in the (Fig. 4) are done on the datasets before the words are converted to a suitable numeric/binary type.

Step 4: Feature scaling (converting words into vectors) is applied on datasets $D_{labeled}$ and $D_{unlabeled}$ by applying TF-IDF. Further the dataset, $D_{labeled}$ is split into training $D_{labeled}$ dataset and test $D_{labeled}$ dataset. Preprocessing the data involves the same steps that were shown in Framework-1.

Step 5.a: Let the number of tuples in the unlabeled dataset be T. For each of these tuples the labels are predicted using a random prediction approach.

Step 5.b.: Train an SVM classifier, x on the tuples labelled in step 5, and classify the labelled tuples.

Step 5.c: Calculate the accuracy and fitness score of the SVM classifier, x.

Step 6. Run the step 5 for N times. This is the completion of one generation. At the end of the generation, the fitness score of N SVM classifiers is obtained.

$$fitness\ score = \{SVM-1, SVM-2, \dots, SVM-N\} \quad (5)$$

Fitness score is a measure of accuracy that the model shows on labelled dataset.

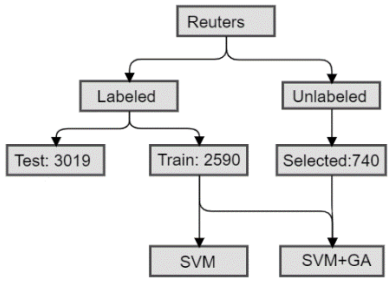

Fig. 6. Tree Representation of Smaller Dataset.

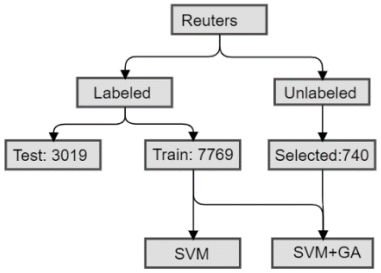

Fig. 7. Tree Representation of Bigger Dataset.

Step 7: After calculating the fitness score of each classifier, the mating pool is created. The purpose of mating pool is to identify the two SVM classifiers that has highest accuracy among N other SVM classifiers

$$fitness\ score = \{SVM-1 > SVM-2 > \cdots SVM-N\} \quad (6)$$

Here the models are arranged in decreasing order of their fitness scores.

Step 8: Now starts the Genetic algorithm operations

Step 8.a: Crossover: The crossover is done on the labels of the tuples generated by the top two classifiers chosen in Step 7. An initial probability, p, for cross over is chosen by the algorithm during runtime. The value of p decides crossover between the results of the two classifiers. If it is lower than the number expected, crossover is performed otherwise the parents are passed on as the off springs (new population) (Fig. 8).

Step 9: Mutation: Randomly change the values of the labels classified by the SVM and this represents the new set of labels for the unlabeled dataset. A mutating variable, p1 acts as a deciding factor to change the values of the tuples (Fig. 9).

Step 10: Step 5 to Step 9 is named as one generation in GA. Check if the number of generations, G is reached. The value of G is decided based on the accuracy of the final classifier generated. If not, then go back to step 5, otherwise move on go step 11.

Step 11: At the end of G generations, all the labelled tuples of the unlabeled dataset is moved to train $D_{labeled}$ dataset.

Step 12: An SVM is trained on the new train $D_{labeled}$ dataset and the performance is measured.

Step 13: Test this classifier on the test $D_{labeled}$ dataset and the performance is measured.

The generated SVM on applying TF-IDF and GA classifies the documents into 90 classes and produces better accuracy.

In the (Fig. 10), the diagrammatic representation of the algorithm used is shown.

Before Cross-over:

Labels of first best SVM	$Te1_1$	$Te1_2$	$Te1_3$	...	$Te1_{369}$	$Te1_{370}$	$Te1_{371}$	$Te1_{372}$	...	$Te1_{740}$
Labels of second Best SVM	$Te2_1$	$Te2_2$	$Te2_3$	...	$Te2_{369}$	$Te2_{370}$	$Te2_{371}$	$Te2_{372}$	...	$Te2_{740}$

After Cross-over:

Labels of first best SVM	$Te1_1$	$Te1_2$	$Te1_3$	...	$Te1_{369}$	$Te2_{370}$	$Te2_{371}$	$Te2_{372}$	...	$Te2_{740}$
Labels of second best SVM	$Te2_1$	$Te2_2$	$Te2_3$	...	$Te2_{369}$	$Te1_{370}$	$Te1_{371}$	$Te1_{372}$	...	$Te1_{740}$

Fig. 8. Cross-over.

Before Mutation	Te_1	Te_2	Te_3	Te_4	Te_5	Te_6	Te_7	...	Te_{739}	Te_{740}
After Mutation	Te_1	Te_2	Te_3	Te_4	Te_5	Te_6	Te_7	...	Te_{739}	Te_{740}

Fig. 9. Mutation.

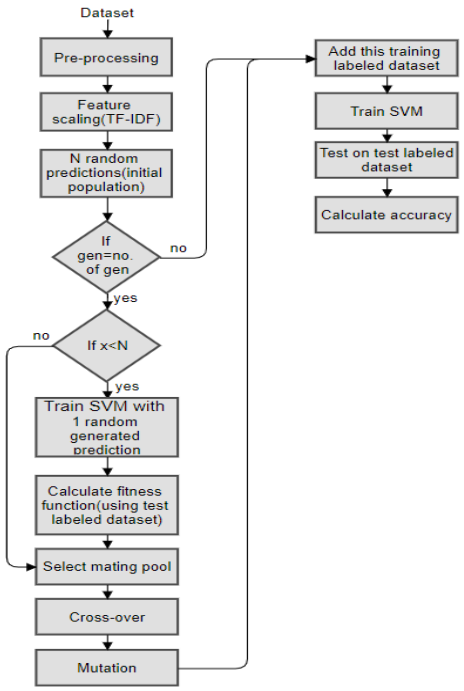

Fig. 10. Framework-2.

IV. Implementation and Results

In this section the performance of frameworks is evaluated based on evaluation metrics:

A. Performance Metrics

In the following section, a brief explain about the performance metrics used is given.

Accuracy and F1-score will be used to compare the performance of the proposed and existing frameworks.

Accuracy can be defined as correctly classified points over total instances in the dataset.

Accuracy= True Positive + True Negative / True Positive + True Negative + False Positive + False Negative (7)

Precision is the ratio of correctly predicted positive instances over total predicted positive instances.

Precision=True Positive / True Positive + False Positive (8)

Recall is the ratio of correctly predicted positive instances over actual positive instances.

Recall= True Positive/True Positive + False Negative (9)

F1-score takes both false positives as well as false negatives into account. For a good F1-score, good precision as well as good recall is needed. Having a model with good F1-score will be better than the one with lower F1-score. F1-score is 2 times the inverse of harmonic mean of Precision and Recall.

F1-score= (2* Precision * Recall)/ (Precision + Recall) (10)

TABLE II. CONFUSION MATRIX FOR FRAMEWORK-1

		Existing Model		*Proposed Model*		*Existing Model*		*Proposed Model*	
		SVM(BNS)		*VC(BNS)*		*SVM(TF-IDF)*		*VC(TF-IDF)*	
		Predicted Class Label							
Actual Class Label		*0(Not Spam)*	*-1(Spam)*	*0(Not Spam)*	*-1(Spam)*	*0(Not Spam)*	*-1(Spam)*	*0(Not Spam)*	*-1(Spam)*
	0(Not spam)	0	0	0	0	130	2	131	2
	-1(Spam)	161	954	149	966	19	964	17	965

B. Confusion Matrix of all the 4 Models BNS vs TF-IDF

The evaluated model's confusion matrix is given in Table II. The spam text is classified into two classes: spam and not spam. The contemporary SVM with BNS and TF-IDF is compared with the proposed Voting Classifier with BNS and TF-IDF.

C. Tabulated Results of Framework-1

The results of the framework-1 are tabulated in Table III. It is found that accuracy score and F1-score of models when TF-IDF is used much better than the models when BNS is used.

Another notable feature is that Voting Classifiers perform better than SVMs. TF-IDF when used with Voting Classifier shows the best results. Therefore, TF-IDF gives better real valued representation of textual data when converted into vectors.

The following figures (Fig. 11 and Fig. 12) show the Accuracy scores and F1-scores of all the 4 models.

1) SVM (RBF kernel) with BNS
2) VC with BNS
3) SVM (RBF kernel) with TF-IDF
4) VC with TF-IDF

TABLE III. PERFORMANCE METRIC VALUES OF THE 4 MODELS

Models/Evaluation Metrics	*Accuracy*	*F1-score*
SVM(BNS)	85.6%	91.2%
VC(BNS)	86.3%	92.8%
SVM(TF-IDF)	98.11%	98.2%
VC(TF-IDF)	98.26%	98.9%

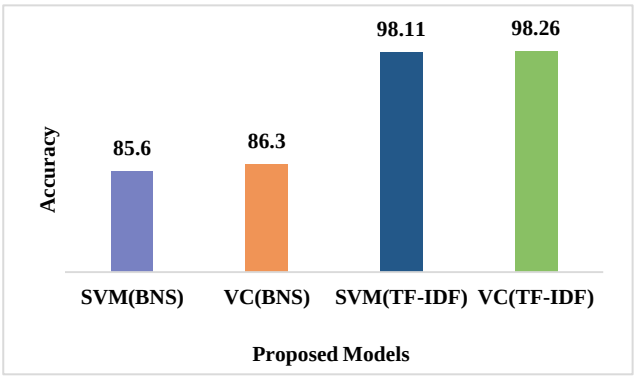

Fig. 11. Accuracy Measures of Framework-1.

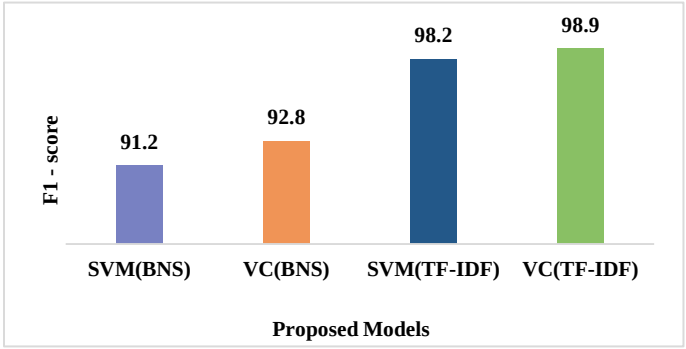

Fig. 12. F1-scores of Framework-1.

TABLE IV. PERFORMANCE METRIC VALUES OF SVM AND SVM+GA OVER SMALLER DATASET, 1- GENERATION AND 3- GENERATIONS

	1-generation		**3-generation**	
Models/Evaluation Metrics	***Accuracy***	***F1-score***	***Accuracy***	***F1-score***
SVM	6.12%	11.59%	6.12%	11.59%
SVM+GA	20.07%	27.98%	36.52%	42.82%

D. Tabulated Results of Framework-2

On applying genetic algorithms over SVM, the performance of the system shows greater increase with small labelled dataset. Accuracy and F1-scores of SVM applied over small labelled dataset (Table IV) are 6.12% and 11.59% whereas SVM+GA are 20.07% and 27.98%. As we can see that when GA is implemented for 1 generation and 3 generation, there is a huge difference between the results of SVM and proposed methods in terms of evaluation metrics. The GA is applied on a smaller dataset to show the increase in the performance.

Over the complete Reuters dataset SVM and SVM + GA is applied and the performance is measured and Tabulated in Table V. It's found that with GA, a substantial increase in accuracy is found, as because GA learns the pattern of allocating labels to the unlabeled data that are much closer to the original labels.

TABLE V. PERFORMANCE METRIC VALUES OF SVM AND SVM+GA OVER COMPLETE DATASET, 1- GENERATION AND 3- GENERATIONS

	1-generation		**3-generation**	
Models/Evaluation Metrics	***Accuracy***	***F1-score***	***Accuracy***	***F1-score***
SVM	86.53%	88.12%	86.53%	88.12%
SVM+GA	90.37%	91.45%	93.88%	96.72%

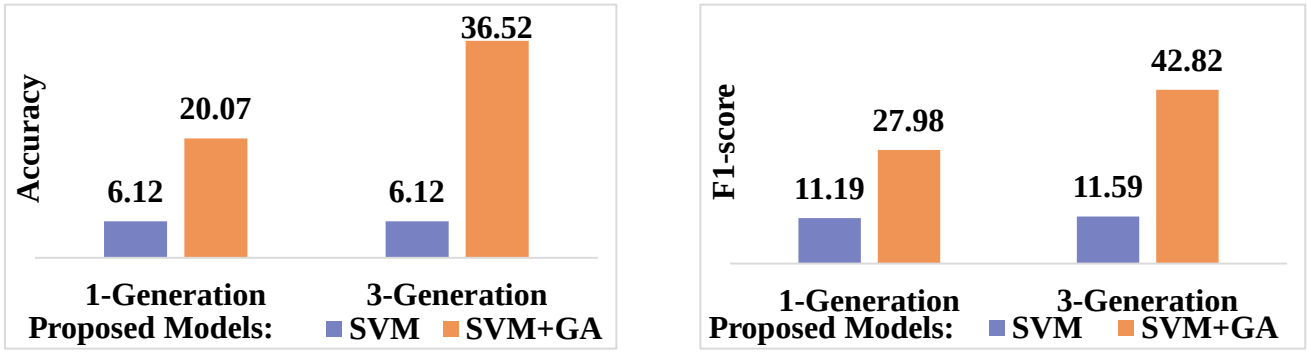

Fig. 13. Accuracy and F1 Score of Measures of Framework-2 with a Small Labelled Dataset.

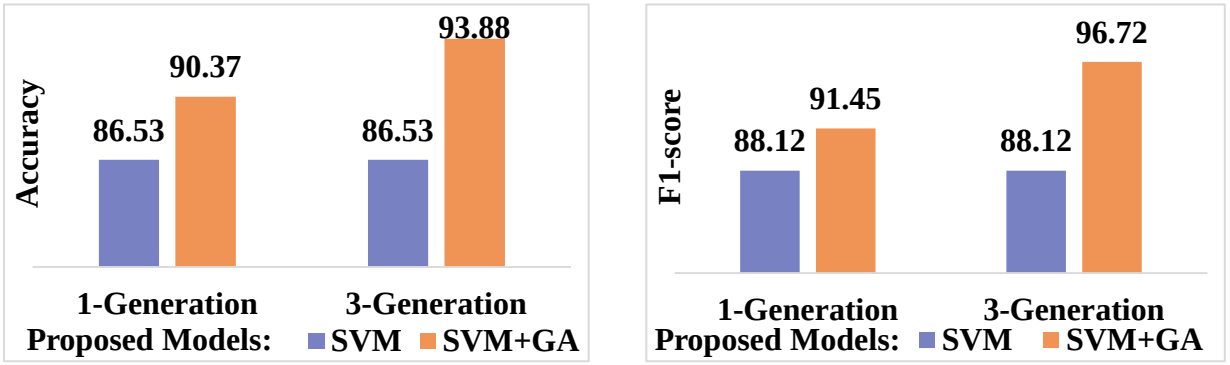

Fig. 14. Accuracy and F1 Score of Measures of Framework-2 with a Large Labelled Dataset.

Following figures (Fig. 13 and Fig. 14) show the graphical representation stating the accuracy and F1- score of SVM and SVM+GA with small and large labelled dataset.

V. Conclusion

This paper proposes two algorithms for binary and multi-class text classification. The two-vector representation of textual data: TF-IDF and BNS is explored and is identified that TF-IDF produces good results than BNS. The contributions are made in two-fold. The first proposed algorithm is the voting classifier that is created with four different classifiers. It is tested on Spam Text classification dataset from Kaggle. From results it is found that the use of TF-IDF improves the accuracy than BNS and the results ensured that Voting Classifier when used along with TF-IDF will give best accuracy (98.26%). The second algorithm classifies the dataset with limited number of labelled training samples. This algorithm applies Genetic Algorithm to generate labels for unlabeled datasets over which SVM is applied. Cross over and Mutation are the genetic operations that are applied in many generations. The results are measured in two folder: with smaller labelled dataset and with larger labelled dataset. With a small labelled dataset, an increase of accuracy from 6.12 % to 20.07% is found with one generation and 6.12 % to 36.52% with three generations on applying SVM and SVM+GA. With a large labelled dataset, an increase of accuracy from 86.53% to 90.37% is found with one generation and 86.53% to 93.88% with three generations on applying SVM and SVM+GA. This proposed algorithm is applied on the Reuters dataset. An increase in generation and on application of GA operations might slow down that process but has a positive side that helps in increasing the accuracy by a good margin. The future scope will be to incorporate genetic algorithm along with neural networks for the task of Text Classification. We also aim to research on using different pre-processing steps that will enhance the model's performance metrics score.

References

[1] Basu, Atreya, Christine Walters, and M. Shepherd. "Support vector machines for text categorization." 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the. IEEE, 2003.

[2] Salazar, Diego Alejandro, Jorge Iván Vélez, and Juan Carlos Salazar. "Comparison between SVM and logistic regression: Which one is better to discriminate?." Revista Colombiana de Estadística 35.SPE2 (2012): 223-237.

[3] Liu, Zhijie, et al. "Study on SVM compared with the other text classification methods." 2010 Second international workshop on education technology and computer science. Vol. 1. IEEE, 2010.

[4] Colas, Fabrice, and Pavel Brazdil. "Comparison of SVM and some older classification algorithms in text classification tasks." IFIP International Conference on Artificial Intelligence in Theory and Practice. Springer, Boston, MA, 2006.

[5] Rajvanshi, Nitin, and K. R. Chowdhary. "Comparison of SVM and Naïve Bayes Text Classification Algorithms using WEKA." International Journal of Engineering Research and 6 (2017): 09.

[6] Alsaleem, Saleh. "Automated Arabic Text Categorization Using SVM and NB." Int. Arab. J. e Technol. 2.2 (2011): 124-128.

[7] Forman, George. "BNS feature scaling: an improved representation over tf-idf for svm text classification." Proceedings of the 17th ACM conference on Information and knowledge management. 2008.

[8] Lan, Man, et al. "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines." Special interest tracks and posters of the 14th international conference on World Wide Web. 2005.

[9] Dadgar, Seyyed Mohammad Hossein, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. "A novel text mining approach based on TF-

IDF and Support Vector Machine for news classification." 2016 IEEE International Conference on Engineering and Technology (ICETECH). IEEE, 2016.

[10] Lilleberg, Joseph, Yun Zhu, and Yanqing Zhang. "Support vector machines and word2vec for text classification with semantic features." 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE, 2015.

[11] Worcester, C. O. "TEXT CLASSIFICATION WITH LEAST SQUARE SUPPORT VECTOR MACHINES AND LATENT SEMANTIC INDEXING."

[12] Mitra, Vikramjit, Chia-Jiu Wang, and Satarupa Banerjee. "A neuro-svm model for text classification using latent semantic indexing." Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.. Vol. 1. IEEE, 2005.

[13] Xu, Zhenqiang, Pengwei Li, and Yunxia Wang. "Text classifier based on an improved SVM decision tree." Physics Procedia 33 (2012): 1986-1991.

[14] Ramaswamy, Srinivasan. "Multiclass text classification a decision tree based SVM approach." CS294 Practical Machine Learning Project (2006).

[15] Demidova, L. A., I. A. Klyueva, and A. N. Pylkin. "Hybrid approach to improving the results of the SVM classification using the random forest algorithm." Procedia Computer Science 150 (2019): 455-461.

[16] Silva-Palacios, Daniel, Cesar Ferri, and María José Ramírez-Quintana. "Improving performance of multiclass classification by inducing class hierarchies." Procedia Computer Science 108 (2017): 1692-1701.

[17] Hao, Pei-Yi, Jung-Hsien Chiang, and Yi-Kun Tu. "Hierarchically SVM classification based on support vector clustering method and its application to document categorization." Expert Systems with applications 33.3 (2007): 627-635.

[18] Yuan, Pingpeng, et al. "MSVM-kNN: Combining SVM and k-NN for Multi-class Text Classification." IEEE international workshop on Semantic Computing and Systems. IEEE, 2008.

[19] Manevitz, Larry M., and Malik Yousef. "One-class SVMs for document classification." Journal of machine Learning research 2.Dec (2001): 139-154.

[20] Ramanathan, Thirumalaimuthu Thirumalaiappan, and Dharmendra Sharma. "Multiple Classification Using SVM Based Multi Knowledge Based System." Procedia computer science 115 (2017): 307-311.