

A Step towards the Improvement in the Performance of Text Classification

Shahid Hussain^{1*}, Muhammad Rafiq Mufti², Muhammad Khalid Sohail³, Humaira Afzal⁴,
Ghufran Ahmad⁵ and Arif Ali Khan⁶

^{1,5} Department of Computer Science, COMSATS University Islamabad, Islamabad

³ Department of Management Science, COMSATS University Islamabad, Islamabad

² Department of Computer Science, COMSATS University Islamabad, Vehari

⁴ Department of Computer Science, Bahauddin Zakariya University, Multan

⁶ College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics

¹shussain@comsats.edu.pk, ²rafiq_mufti@ciitvehari.edu.pk, ³khalid_sohail@comsats.edu.pk,

⁴humairaafzal@bzu.edu.pk, ⁵ghufran.ahmed@ieee.org, ⁶arif.khan@nuaa.edu.cn

*Corresponding author: Shahid Hussain

*Received March 12, 2018; revised August 1, 2018; accepted October 17, 2018;
published April 30, 2019*

Abstract

The performance of text classification is highly related to the feature selection methods. Usually, two tasks are performed when a feature selection method is applied to construct a feature set; 1) assign score to each feature and 2) select the top-N features. The selection of top-N features in the existing filter-based feature selection methods is biased by their discriminative power and the empirical process which is followed to determine the value of N. In order to improve the text classification performance by presenting a more illustrative feature set, we present an approach via a potent representation learning technique, namely DBN (Deep Belief Network). This algorithm learns via the semantic illustration of documents and uses feature vectors for their formulation. The nodes, iteration, and a number of hidden layers are the main parameters of DBN, which can tune to improve the classifier's performance. The results of experiments indicate the effectiveness of the proposed method to increase the classification performance and aid developers to make effective decisions in certain domains.

Keywords: Feature Set, Text Classification, Classifiers, Deep Learning, Performance.

1. Introduction

Text classification approach has been deployed in certain domains to automate the systems in order to decrease the computational time and cost. The goal of such systems is to categorize the texts in to proper classes based on the content. For example, filtering of spam emails [1], classification of web page [2], analysis of sentiments [3], identification of authors [4], and design pattern classification and selection [5, 6]. In these domains, the use of a large number of features affects the classification decisions and computation time of the classifiers used in an automated system [7]. In order to reduce the adverse effect of inoperable features in text categorization, filter-based (rather than embedded and wrapper) feature selection techniques are applied. [8]. Though numerous filter-based methods such as DF(Document Frequency [9], IGI(Improved Gini Index) [10], Correlation, GR (Gain Ratio), IG (Information Gain) [11], and Distinguish Feature Selector (DFS) [12] have been introduced and applied in the text classification domains [7, 13]. However, due to variation in the discriminative power of filter-based feature selection techniques and their constructed feature set, we still required a systematic way to construct a more representative feature set. Researchers have successfully implemented deep learning algorithms in certain domains such as fault localization [14], organization of design patterns according to expert opinion [15], extraction of semantic features from the source code, digit recognition [16], code suggestion via modeling the programming languages [17], use of structural information of program, and use of source code of programs to extract features for defects prediction [18, 19]. Due to the successful implementation of deep learning algorithms in certain areas [20] and construction of a more demonstrative feature set, we leverage a powerful deep learning technique. Like Hussain et al. [15], we leverage an influential and representative algorithm, namely DBN [21] to cram features from the constructed VSM (Vector Space Model). The DBN algorithms aims to aid in reconstruction of highly likely training data. Deep Belief Network (DBN) is a multi-level neural network, which can aid to obtain the high level abstraction of input data by constructing a deep architecture. A Deep Belief Network (DBN) is consists of one visible and numerous hidden layers. Each layer is the combination of numerous stochastic neurons and is fully connected with adjacent layers. We accomplish few pre-processing tasks such as removal of stopwords and word stemming to realm the data in feature vector form with their frequency. The input of DBN algorithm is constructed through these feature vectors.

In this paper, we accomplished several experiments. In each experiment, three widely used classifiers are used to assess the efficacy of the proposed method in terms of improving their classification decisions. The major contributions of the proposed work are.

- We used the DBN algorithm to retrieve the influential features. The DBN learn these features from the document vocabulary in order to increase the classifier's performance.
- We appraise the capability of the proposed method by comparing with the state of the art filter-based feature selection techniques in terms of construction of more representative feature set. .
- We employed the proposed method with three well-known classifiers and three benchmark datasets in the context of text classification and measure the efficacy of the proposed method.

Several weighting schemes are used to identify and assign weights to features and rank them accordingly. Such as, Binary, TF (Term Frequency), Entropy, TFIDF (Term Frequency Inverse Document Frequency), and LTC (Length Term Collection) are the well-known weighting schemes [22]. These weighting schemes are used to assign weights to each feature

and ordered them accordingly.

The remaining article is categorized in eight units. In section 2, we concise the allied work in the context of feature selection techniques and implication of deep learning in the domain of software engineering. In section 3, we describe the working process of the proposed method. In section 4, we present a toy case study. In section 5, we present the experimental procedure. In section 6, we discussed the experimental results and discussed consequences. Moreover, in section 7, we concise the assessment results. In section 8 and section 9, we present the few threats of proposed method and the conclusion of proposed work respectively.

2. Related Work

We summarized the related work to highlight the use of existing filter-based techniques and implication of deep learning approach (for feature set construction) in the context of increase in the performance of text classification based automated system.

2.1. Feature Selection Methods

The feature selection techniques are used to decrease the adverse effect of feature selection techniques. The existing feature selection techniques can be categorized in three schemes such as filter-based, wrapper, and embedded. The feature selection methods of last two schemes (i.e. wrapper and embedded) need a learning model [7]. In their study, Ogura et al. [23] describe the characteristics of filter-based methods and group them as one-sided or two-sided. In case of one-sided methods, each feature is assigned several scores depends on the number of class labels. However, these scores can be group with respect to the membership class score (i.e. $\text{Score} \geq 0$) and non-membership classes (i.e. $\text{Score} < 0$). The Odd Ratio (OR) is the common example of one-sided method. The aim of OR is to compute the membership and non-membership scores of each feature by normalizing its nominator and denominator values. Moreover, in case of two-sided methods, a single score (based on the amalgamation of non-membership and membership) is associated to each feature. The IGI, DF, and IG are the common examples of two-sided feature selection methods [6, 13]. Similarly, in their study, Tasci and Gungor analysed certain filter-based techniques in terms of assigning single or multi class based score to each feature and group them as local and global [24]. The classifier's performance in the context of text classification can be improved in terms of handling the class imbalances issue. In this regard, Gunal empirically investigate the progress in the classification performance of genetic algorithm and filter based methods [25]. Finally, the authors informed that combine filter methods outperform than a single filter method.

Recently, Uysal present an IGFSS (Improved Global Feature Selection Scheme) scheme to construct more demonstrative feature set and reported the improvement in the classification decision of classifiers. The aim of IGFSS is construct a feature set by evenly class-wise distribution of features [7].

Though, we observed several methods from the literature, however, in the context of text classification, feature selection is still an enduring issue for research community. This thing motivates us to present a new method for the construction of a more demonstrative feature set to increase the classifier's performance.

Table 1. Summary of existing related work

Study	Context	Objective of Study
S. Wang et al. [18]	Prediction of Defects	Authors used the DBN (Deep Belief Network) a deep learning algorithm to learn the semantic features. These features are extracted from the token vectors. These vectors are retrieved from the Abstract Syntax Tree of a program. The DBN is introducing with aim to build the correct prediction models.
Yang et al. [19]	Prediction of Defects	Yang et al. used the deep learning to harvest the features from the list of features and employed them to predict the buggy commit.
Lam et al. [14]	Localization of Fault	Lam et al. target the certain information retrieval methods and deep learning algorithms and investigate the improvement in fault localization.
White et al. [17]	Code relevant Suggestions	White et al. employed the deep learning to exemplary the languages for code suggestion.
Pascanu et al. [26]	Classification of Malware	Pascanu et al. present a method similar to a model of natural language. The aim of propose method is to extract the features by learning the language malware via executing the instructions.
Hussain et al [15]	Design Pattern Organization	In this work, the authors have used the DBN to construct a more representative feature set in order to organize the design patterns with respect to domain expert opinions.
Liu [27]	Classification decisions	In this work, the author has leverage the DBN algorithm to improve the classification decision of Support Vector Machine (SVM) for Chinese text.

2.2. Deep learning in Software Engineering (SE) domain

In the Software Engineering (SE) domain, research community has introduced and adopted several deep learning algorithms for certain tasks. In this regard, the related work has been summarized and shown in [Table 1](#), which presents its importance.

In a recent study Hussain et al. [15] have leveraged the DBN (Deep Belief Network), a deep learning algorithm for construction of a more demonstrative feature set. In their study, Hussain et al. [15] accomplished three case studies to describe the effectiveness of the proposed method for the organization of software design patterns in the perspective of different domains. The organization of software design patterns is evaluated with respect to classification schemes of domain experts. Liu [27] leverage the DBN algorithm and introduce a new text classification method to improve the classification decision of Support Vector Machine (SVM) for the Chinese text. The aim of Liu [27] proposed study is to improve the F1 measure by incorporating DBN to develop high-level abstraction of text for classification decision via SVM. The readers who are more interested to know about the applications of deep learning approach can study the Bengio et al. work [20].

3. Proposed method

In order to address the biasness of existing filter-based methods, we look out the capability of leveraged algorithm in the domain of construction of a more demonstrative feature set to increase the classifier's performance. The study of proposed method is structured into three phases namely Pre-processing (Phase 1), Construction of Feature Set (Phase 2), and Text classification (Phase 3).

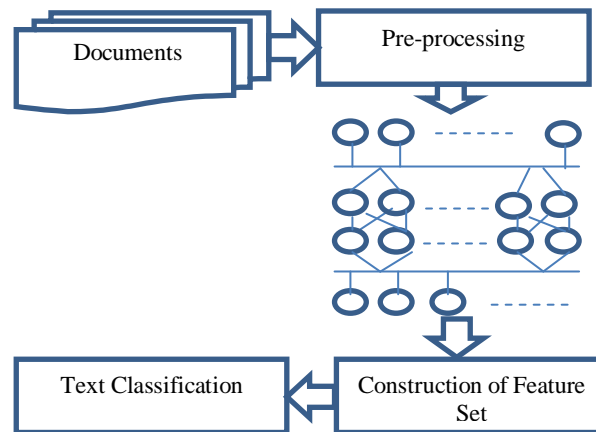


Fig. 1. Layout of the proposed method

3.1. Preprocessing

In the pre-processing phase, three tasks are used to transform documents into strings. Further, these tasks aid to formulate strings into feature vector form. The aim of the first task is used to remove the more frequent stop words. Such words carry no information such as Pronouns, Conjunction, and Preposition. The aim of second task is to stem the group of related words which have same conceptual meaning. Through, through this task numerous words are removed. Porter's stemmer¹ is a widely used word stemming algorithm whose aim is transform English words into their stem using a set of rules. Finally, the aim of third task is describe the each feature vector into an integer vector form as a prerequisite for DBN algorithm. Though, research community has reported certain weighting methods such as Entropy, TF, TFIDF, Binary, LTC, and TFC. However, in this paper, we employ the TF for construction of integer vectors. The size of integer vectors is same with respect to prerequisite inputs of DBN algorithm.

3.2. Feature Set Construction

In this phase, a feature set is constructed for appropriate classification of documents using the DBN algorithm. The goal of DBN algorithm is to learn from given input (feature vectors) through a multi-level neural network. Further, it reconstructs the highly likely input data. The DBN has a three layer architecture, which are 1) one input, 2) N hidden, and 3) one output (i.e. Top) layer. Each layer is include a set of nodes. The data complexity can be tune by selecting the number of hidden layers and nodes in each layer. The joint distribution of input and hidden layers can be described through equation 1.

¹<http://snowball.tartarus.org/algorithms/porter/stemmer.html>

$$P(v, h^1, \dots, h^l) = P(v | h^1) \left(\prod_{k=1}^l P(h^k | h^{k+1}) \right) \quad (1)$$

Where the terms v , l , and h^k referred as feature vector of input layer, total number of hidden layers, feature vector of k^{th} layer respectively. The $P(h^k | h^{k+1})$ is used to described the conditional distribution of two adjacent layers (for example k and $k+1$). This distribution is calculated from Restricted Boltzmann Machines (RBM). In case of node, DBN learns probabilities from any current node to the upper levels nodes. The backward validation is performed by DBN to reconstruct the input feature set by tuning the weights between nodes across the layers. The input visible node of $(j+1)^{\text{th}}$ RBM are retrieved from hidden node the j^{th} RBM. The Energy E for a visible layer v and hidden layer h is computed through equation 2.

$$E(v, h) = -\sum_{i=1}^{n^v} b_i^v v_i - \sum_{j=1}^{n^h} b_j^h h_j - \sum_{i=1}^{n^v} \sum_{j=1}^{n^h} v_i h_j w_{ij} \quad (2)$$

Where v_i and h_j are the i^{th} and j^{th} node of v and h . The b_i^v and b_j^h are the biases associated with nodes v_i and h_j respectively. w_{ij} is the weight between v_i and h_j . The relationship between Energy E and probability of generated data is proportional represented via equation 3 and equation 4.

$$\partial E(v, h) / \partial w_{ij} = -v_i h_j \quad (3)$$

$$P(v) = \sum_h e^{-E(v, h)} / \sum_{u, g} e^{-E(u, g)} \quad (4)$$

The output h_j of a visible unit v for one RBM is represented via equation 5.

$$P(h_j = 1 | v) = \sigma(b_j^h + \sum_{i=1}^{n^v} w_{ij} v_i) \quad (5)$$

Where $\sigma(x) = 1 / (1 + e^{-x})$. The current state of the hidden node is used to derive a new visible node using equation 6

$$P(v_i = 1 | h) = \sigma(b_i^v + \sum_{j=1}^{n^h} w_{ij} h_j) \quad (6)$$

3.3. Text Classification

Though, in text classification based automated system, several supervised learning techniques are used. However, no single outperformed classifier is reported for all problems [28]. So, it is endorsed to find the outperform classifier for any new problem. Moreover, in the text categorization domain, research community have reported the better performance of Naïve Bayes, Support Vector Machine (SVM), and C4.5 Decision Tree [7]. In this paper, we considered these classifiers to assess the efficacy of the proposed method.

4. Toy Case Study

The aim of toy case study is to present the analysis and working way of proposed method. The corpus of Toy case study consists of eight documents, 11 unique features and 3 classes. The description of documents is shown in [Table 2](#).

Table 2. Document's description for Toy case study

Documents	Content	Class
Document 1	Rose Aconitum Sunflower Angelonia	C1
Document 2	Rose Aconitum Alchemilla Jasmine Alyssum	C2
Document 3	Jasmine Rose Sunflower Alyssum	C2
Document 4	Alchemilla Anemone Amaranthus Anthurium	C3
Document 5	Anemone Alchemilla Amaranthus Lily	C3
Document 6	Aconitum Lily Alchemilla Amaranthus Sunflower	C3
Document 7	Anthurium Lily Anemone Sunflower	C3
Document 8	Aconitum Rose Alchemilla Anemone Amaranthus Alyssum Angelonia Anthurium Sunflower Jasmine Lily	C2

Though, selection of top-N feature remains an issue to build a more demonstrative feature set. However, the ranking of features with respect to discriminative power of feature selection can help to construct a feature set.

Table 3. Document's description for Toy case study

Features	Frequency	Features Ranking with respect to feature selection methods				
		DF	IG	GI	GR	Proposed method (DBN)
Rose	4	2	1	2	1	5
Aconitum	4	2	3	4	3	3
Jasmine	3	3	1	1	1	2
Alchemilla	5	1	2	1	2	6
Sunflower	5	1	4	2	4	5
Lily	4	2	2	2	2	7
Alyssum	3	3	1	5	2	2
Amaranthus	4	2	2	1	2	7
Anemone	4	2	2	2	2	4
Anthurium	3	3	4	3	4	4
Angelonia	2	4	2	2	3	1

The pre-processing activities (i.e. Section 3.1) on corpus (i.e. [Table 2](#)) are applied. The list of unique features with their frequencies and ranking with respect to discriminative power of feature selection method is shown in [Table 3](#).

The result of ranking of unique features of Toy example can help to construct a feature set based on the top-N values (i.e. the value of N can be selected in range 1 to 11). Since Document Frequency (DF) feature selection method depends on the frequency of a unique feature in whole corpus. Consequently, the frequency of features can be used to rank the features. Such as, the features “Alchemilla” and “Sunflower” are ranked first based on their highest frequencies. Similarly, the features are ranked according to discriminative power of IG, GR, GI and proposed method (DBN). The feature set with N=5 for each selected method is shown in [Table 4](#).

Table 4. Feature sets with top N=5

Feature Selection Method	List of features in Feature set with top N=5
DF	Alchemilla, Sunflower, Rose, Aconitum, Lily
IG	Rose, Jasmine, Amaranthus, Alchemilla, Lily
GI	Jasmine, Alchemilla, Amaranthus, Rose, Sunflower
GR	Rose, Jasmine, Alchemilla, Lily, Alyssum
Proposed Method	Angelonia, Jasmine, Alyssum, Alyssum, Aconitum

There are two main observations from the result of [Table 4](#) are; 1) the feature set varies with respect to included features and 2) the number of orders of features in a constructed feature set.

5. Experiment Setup

In this section, we performed several experiments to examine the efficacy of the proposed method against the performance of four global filter-based methods namely DF [\[9\]](#), IGI [\[10\]](#), GR and IG [\[7, 11\]](#).

5.1. Dataset

We used three widely-used dataset with varying characteristics named Reuters-21578 ModApte Split², WebKB³, and Classic3⁴. The number of classes of documents and the sample size of each dataset is shown in [Table 5](#).

²<https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>

³<http://www.cs.cmu.edu/~webkb/>

⁴<http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/>

Table 5. Summary of Datasets

Dataset	Total Class Labels	Sample Size
Reuters	12	7193
Classic3	3	2722
WebKB	4	2937

5.2. Evaluation Criteria

In case of text classification, Micro-F1 and Macro-F1 are two well-known measures used to assess the performance of classifiers without or with class discrimination [7]. As, in this paper, our aim to formulate a more illustrative feature set without class discrimination, consequently, we used micro-F1 (shown in equation 9, which incorporate two widely used performance measure precision (equation 7) and recall (equation 8) for the classification decisions.

$$P = \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C (TP_c + FP_c)}, \quad \text{Micro-Averaging} \quad (7)$$

$$R = \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C (TP_c + FN_c)}, \quad \text{Micro-Averaging} \quad (8)$$

$$Micro - F1 = \frac{2 * P * R}{P + R} \quad (9)$$

5.3. Experiment Procedure for Construction of Feature Set

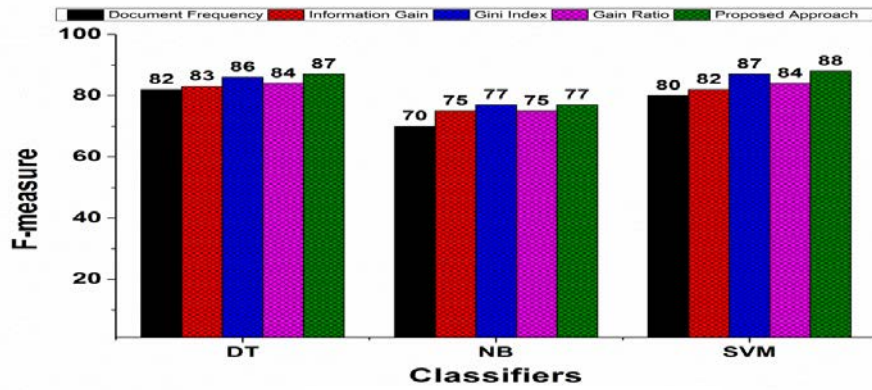
In the proposed method, we used DBN algorithm to build a demonstrative feature set to improve the classifier's performance. We used Darch-CRAN package of R (for statistical computing) to perform experiments. We used DBN algorithm with default parameters (i.e. Nodes in each layer, Number of hidden layers, and iteration) for each dataset. But, the parameter's values can be tune with respect to data complexity in order to obtain more effective outcomes. Certain optimization techniques are required to find settings that clue to high predictive accuracy and the selection of best tuning technique depends on the hyperparameter configurations, and their complex interactions [29, 30]. We used 10-fold cross validation [31] in each experiment and compare the classifier's performance on the constructed feature set of the proposed method and existing filter-based techniques. We keep the number of filtered features is same as we obtain from the proposed method to preserve consistency.

6. Results and Discussion

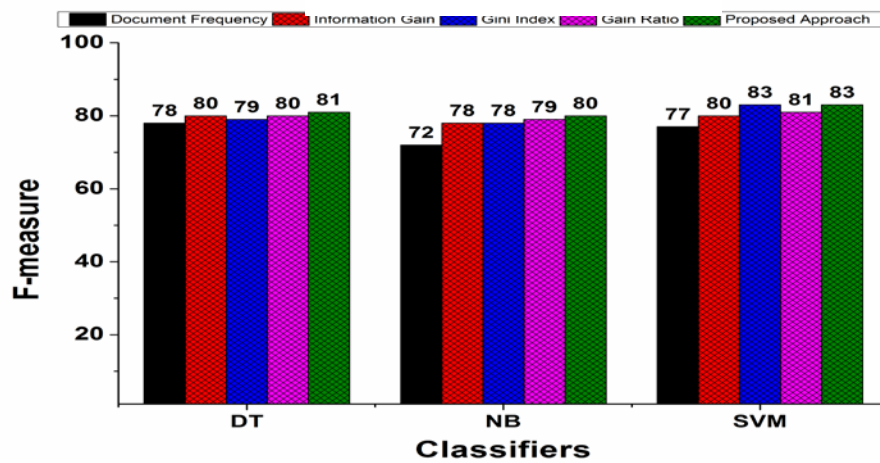
In order to assess the efficacy of the proposed method, we performed three experiments. In each experiment, we equate the efficacy of the proposed method with existing global filter-based methods in term of improvement in classifier's performance. We coined the term classifier instance⁵ to present an association between classifier and constructed feature via a feature selection method. For example, five instances (e.g. NB+IG describe the association between NB and feature set constructed through IG) of Naïve Bayes are created with five

⁵One instance of a classifier is referred to its use with one feature set. For example, in case of SVM, five instances of SVM are created with five feature sets constructed through applied feature selection methods under study

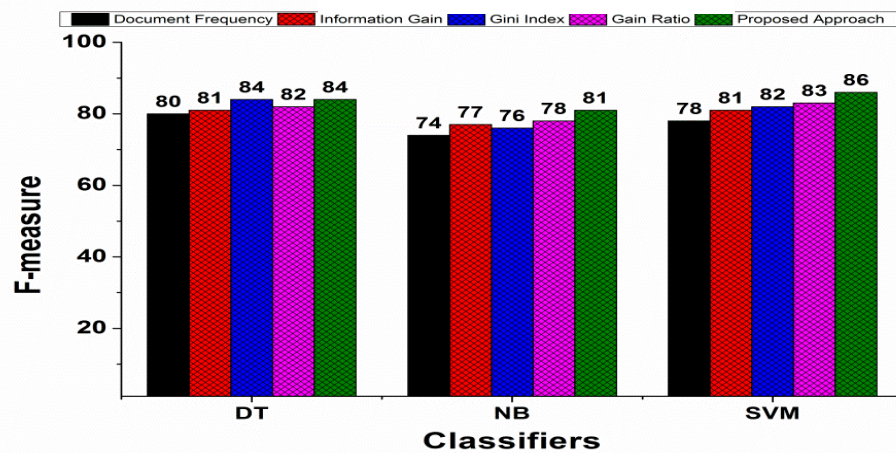
feature sets constructed in the experiment. We consider top 100 ranked features to construct a feature set in each experiment. We used the evaluation criterion describe in the section 5.2 and summarized the results in [Table 6](#) and [Fig. 2](#).



a) Reuters Dataset



b) Classic3 Dataset



c) Web KB Dataset

Fig. 2. Classifier's Performance on a) Reuters b) Classic3 and c) Web KB datasets with feature sets

The main consequences of the experimental results are:

- The performance of classifiers (in term of micro F-measure) on the constructed feature set via the proposed method indicates its applicability and effectiveness.
- In all experiments, on the constructed feature set via the proposed method, the performance of SVM remains better as compared to Naïve Bayes and Decision Tree with a minor difference. For example, in [Fig. 2-a](#), the SVM (88%) outperform than NB (77%) and DT (87%).
- From the comparative assessment, we concluded some outcomes. Firstly, we observed improvement in classifier's performance with a minor difference on feature sets constructed via the proposed method and existing filter-based methods. For example, in [Fig. 2-a](#), the performance of SVM on the feature set of Gini Index and proposed method is reported with minor differences. Secondly, we cannot observe difference in classifier's performance on the feature set of Gini Index and the proposed method. For example, in the [Fig. 2-a](#), the performance of Naïve Bayes is same on both feature sets.
- In some cases, we cannot observe the better performance of classifiers on the feature set of the proposed method. For example, in the [Fig. 2-a](#), the performance of Naïve Bayes on the constructed feature set via the proposed method is not significant (i.e. F-measure =77).

Table 6. Performance evaluation of classifiers in terms of Recall(R), Precision (P), and F-measure

Dataset	Outperformed Classifier	Feature Selection Method	Recall (R) %	Precision (P) %	F-Measure %
Reuters	SVM	Document Frequency	78	82	80
		Information Gain	79	85	82
		Gini Index	84	90	87
		Gain Ratio	82	86	84
		Proposed method (DBN)	82	94	88
Classic3	SVM	Document Frequency	75	79	77
		Information Gain	74	86	80
		Gini Index	79	87	83
		Gain Ratio	77	85	81
		Proposed method (DBN)	75	91	83
Web KB	SVM	Document Frequency	74	82	78
		Information Gain	78	84	81
		Gini Index	80	84	82
		Gain Ratio	78	88	83
		Proposed method (DBN)	79	93	86

The implications of experimental work indicate that there is not a significant difference in the mean F-measure value (or F1 score) of a classifier (with each constructed feature set) across the datasets. For example, there is no significant difference in performance of SVM ([Fig. 2.a-c](#)). Consequently, we applied non-parametric tests 1) Friedman's Test, and 2) post-hoc Nemenyi tests on the F-measure values to determine the significant difference and rank the outperform classifier with the feature set constructed through the DF, IG, GI, and GR. Firstly, Friedman's Test is applied on instances of a classifiers⁵ with three datasets to achieve the

chi-square and p-value shown in [Table 7](#).

The Friedman's Test chi-square value for five instances of each classifier NB, DT, and SVM is greater than the critical value 8.53 with the degree of freedom (df) as 3. Consequently, there is a significant difference between F-measure values. We apply post-hoc Nemenyi and ANOM (Analysis of Means) tests to describe the significant difference between classifier's instances with $\alpha = 0.05$ to reject the null hypothesis (H_0 = All instances of a classifier evenly perform). The results of the [Fig. 3](#) described the ranking of an outperform classifier instance of NB, DT and SVM respectively. From the results of the [Table 4](#) and [Fig. 3](#), we can conclude the significant improvement in the classification performance when proposed method is used to construct the feature set.

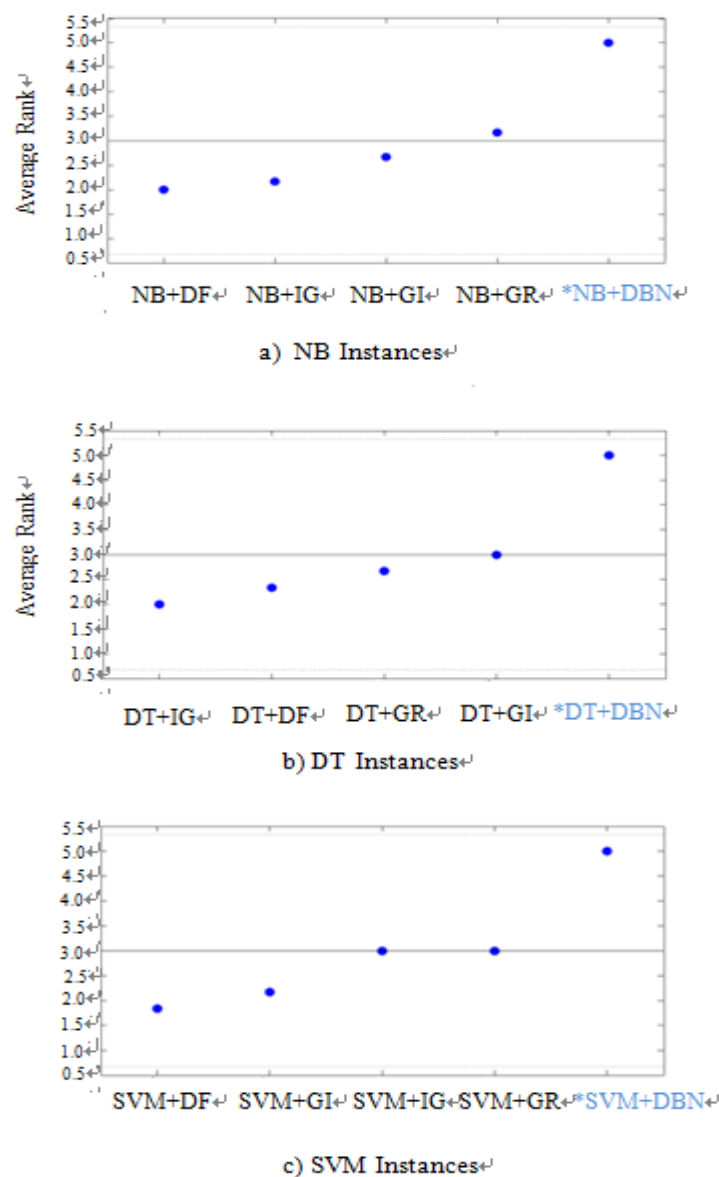
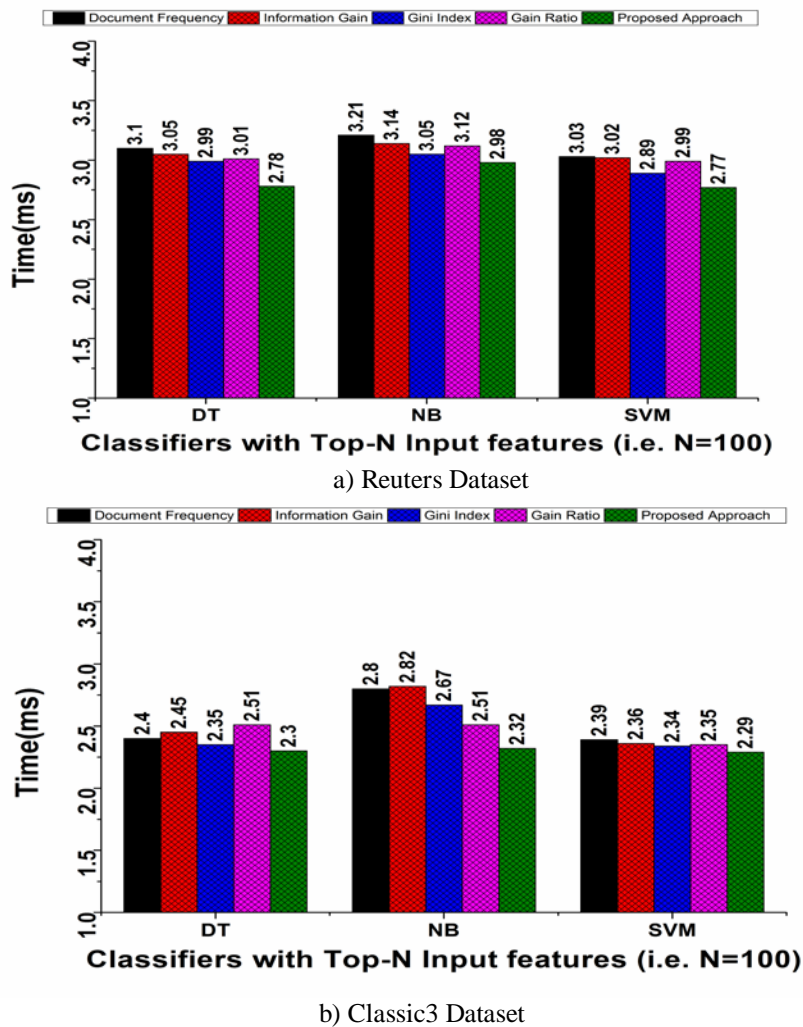


Fig. 3. ANOM Tests Results for Ranking of Instances of a) NB b) DT and c) SVM Classifiers

Table 7. Friedman's Test Results (with F-value) For All Instances of DT (Decision Tree), NB (Naïve Bayes), and SVM(Support Vector Machine) Classifiers

Dataset	Chi-Square	P-Value
DT	9.82	0.04
NB	11.32	0.00
SVM	11.76	0.02

Subsequently, experiments are performed to look for running time of classifier's training and testing. Since 10-fold cross validation is applied so time is recorded from the start to end of its execution. For each dataset, we evaluate the running time of each classifier to present its performance. Since only three datasets are used a benchmark of this study, consequently, we present the running time via bar charts shown in [Fig. 4](#).



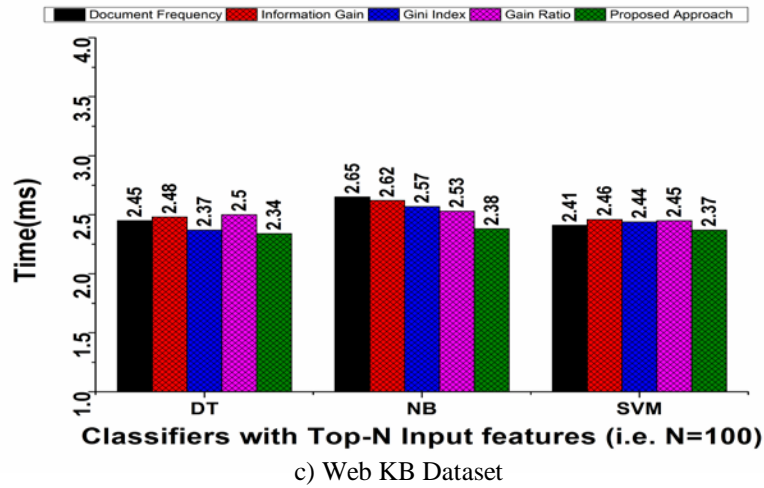


Fig. 4. Runtime analysis of classifiers on a) Reuters b) Classic3 and c) WebKB datasets with feature sets

7. Evaluation Summary

The main aim of the proposed method is to use a widely used deep learning algorithm namely DBN to construct a more demonstrative feature set and assess its influence on the classifier's performance used in the proposed study. In each experiment, three supervised learning techniques are applied to assess the efficacy of the proposed method. The proposed method is employed within the context of three widely used datasets. We observe significant performance of SVM as compared to other classifiers. The main implications of the proposed study are:

- On Reuters Dataset, classification decisions of outperformed classifier SVM with proposed method (i.e. F-measure=80) are 10% improve in terms of F-measure as compared to the classification decision of SVM with Document Frequency feature selection method (F-measure=76). However, we observe 1.2% improvement when SVM is used with Information Gain feature selection method.
- On Classic3 Dataset, classification decisions of outperformed classifier SVM with proposed method (i.e. F-measure=83) are 7.79% improve in terms of F-measure as compared to the classification decision of SVM with Document Frequency feature selection method (F-measure=77). However, we observe very little improvement when SVM is used with Gini Index feature selection method.
- On Web KB Dataset, classification decisions of outperformed classifier SVM with proposed method (i.e. F-measure=86) are 10.25% improve in terms of F-measure as compared to the classification decision of SVM with Document Frequency feature selection method (F-measure=78). However, we observe 3.61% improvement when SVM is used with Information Gain feature selection method.
- We observed the variations in the effect of feature selection techniques on the performance of learners which might be depend on the dataset characteristics. For example, the increase in the performance of outperforms classifier SVM is 10%, 7.79%, and 13.42% for Reuters, Classic3, and Web KB datasets.

8. Threats to Validity

There are few threats in our study. The first threat is to make the generalization of results reported from the experiments. Since, we used a limited number of datasets and classifiers. Consequently, we cannot generalize the experimental results. The proposed study can be replicated with a large number of classifiers and datasets to generalize the results. The second threat is related to the adjustment of effectiveness of DBN algorithm through tuning its default set parameters. Selection of better optimization technique may help to present the best hyperparameter configurations to improve the predictive accuracy. The third threat is related to the usage of performance measure to evaluate the efficacy of the proposed method. The macro-F1 (rather than micro-F1) measure can be used to assess the efficacy of the proposed method for class discrimination. The fourth threat is related to selection of top-N (i.e. $N=100$) features. The results of all experiments are reported with top-N features. Consequently, the reported results may be altered for any other value of N .

9. Conclusion

The proposed method yields promising results which indicate its efficacy in terms of construction of a demonstrative feature set using DBN (Deep Belief Network) algorithm. The construction of feature sets on the base of semantic of words across the documents can aid to increase the classifier's performance in an automated system. We used DBN algorithm for construction of a feature set to present an increase in the classifier's performance. We assess the efficacy of the proposed method with three benchmark datasets and three widely used classifiers DT, NB and SVM. The results of experiments depict the efficacy of the proposed method to improve the classification decisions in an automated system. However, the efficacy of proposed method can be improved by tuning the DBN algorithm's parameters. In the future work, we will analyse the applicability of the proposed method to knob the multi-class problem via tuning parameters.

Acknowledgement

This research is supported by the Higher Education Commission (HEC), Pakistan funds (Project No. 1936/SRGP/R&D/HEC/2018 and 1933/SRGP/R&D/HEC/2018).

References

- [1] I. Idris and A. Selamat, "Improved Email Spam Detection Model With Negative Selection Algorithm And Particles Warm Optimization," *Applied Soft Computing*, 22, p. 11-27, 2014. [Article \(CrossRef Link\)](#)
- [2] E. Sarac and S.A. Ozel, "An Ant Colony Optimization Based Feature Selection For Web Page Classification," *The Scientific World Journal*, p. 1-16, 2014. [Article \(CrossRef Link\)](#)
- [3] W. Medhat, A. Hassan, and H. Korashy, "Sentiment Analysis Algorithms and Applications: A Survey," *Ain Shams Engineering Journal*, 5, p. 1093-1113, 2014. [Article \(CrossRef Link\)](#)
- [4] C. Zhang, X. Wu, Z. Niu and W. Ding, "Authorship Identification From Unstructured Texts," *Knowledge Based Systems*, 66, p. 99-111, 2014. [Article \(CrossRef Link\)](#)
- [5] S. Hussain, J. Keung, A.A. Khan, and K.E. Bennin, "A Methodology to Automate the Selection of Design Patterns," in *Proc. of International Conference on Computers, Software and Application*, IEEE, 2016. [Article \(CrossRef Link\)](#)

- [6] S. Hussain, J. Keung, and A.A. Khan, "Software design patterns classification and selection using text categorization approach," *Applied Soft Computing*, 58, p. 225-244, 2017. [Article \(CrossRef Link\)](#)
- [7] A. K. Uysal, "An Improved Global Feature Selection Scheme for Text Classification," *Expert System with Applications*, 43, p. 82-92, 2016. [Article \(CrossRef Link\)](#)
- [8] H. Uguz, "A Two-Stage Feature Selection Method For Text Classification By Using Information Gain, Principal Component Analysis and Genetic Algorithm," *Knowledge-Based Systems*, 24 (7), p. 1024-1032, 2011. [Article \(CrossRef Link\)](#)
- [9] Y. Yang, and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in *Proc. of the 14th International Conference on Machine Learning*, p. 412-420. 1997. [Article \(CrossRef Link\)](#)
- [10] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A Novel Feature Selection Algorithm For Text Categorization," *Expert Systems with Applications*, 33 (1), p. 1-5, 2007. [Article \(CrossRef Link\)](#)
- [11] C. Lee, and G.G. Lee, "Information Gain And Divergence-Based Feature Selection for Machine Learning-Based Text Categorization," *Information Processing and Management*, 42 (1), p. 155-165, 2006. [Article \(CrossRef Link\)](#)
- [12] A. K. Uysal and S. Gunal, "A Novel Probabilistic Feature Selection Method for Text Classification," *Knowledge-Based Systems*, 36, p. 226-235, 2012. [Article \(CrossRef Link\)](#)
- [13] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *Journal of Machine Learning Research*, 3, p. 1289-1305, 2003.
- [14] A. Lam, A. Nguyen, H. Nguyen, and T. Nguyen. "Combining deep learning with information retrieval to localize buggy files for bug reports," in *Proc. of 30th IEEE ASE Conference*, pages 476-481, 2015. [Article \(CrossRef Link\)](#)
- [15] S. Hussain. et al, "Implications of Deep Learning for the Automation of Design Patterns Organization," *Journal of Parallel and Distributed Computing*, 2017. [Article \(CrossRef Link\)](#)
- [16] G. E. Hinton, R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, 313(5786), p. 504-507, 2006. [Article \(CrossRef Link\)](#)
- [17] M. White, C. Vendome, M. L. Vasquez, and D. Poshyvanyk. "Toward deep learning software repositories," in *Proc. of MSR'15*, p. 334-345, 2015. [Article \(CrossRef Link\)](#)
- [18] S. Wang, T. Liu, and L. Tan, "Automatically Learning Semantic Features for Defect Prediction," in *Proc. of IEEE International Conference on Software Engineering (ICSE)*, 2016. [Article \(CrossRef Link\)](#)
- [19] X. Yang, D. Lo, X. xia, Y. Zhang, and J. Sun., "Deep learning for just-in-time defect prediction," in *Proc. of QRS*, p. 17-26, 2015. [Article \(CrossRef Link\)](#)
- [20] T. Bengio, A. Courville., P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 8, p. 1798-1828, 2013. [Article \(CrossRef Link\)](#)
- [21] G. E. Hinton, S. Osindero, and Y.-W. Teh. "A fast learning Algorithm for deep belief nets," *Neural computation*, 18(7), p. 1527-1554, 2006. [Article \(CrossRef Link\)](#)
- [22] A. Hotho, A. Nurnberger, and G. Paab, "A Brief Survey of Text Mining," *Journal for Computational Linguistics and Language Technology*, 20, p. 19-62, 2005.
- [23] H. Ogura, H. Amano and M. Kondo, "Distinctive Characteristics of a Metric using Deviation from Poisson for Feature Selection," *Journal of Expert Systems with Applications*, 37, p. 2273-2281, 2010. [Article \(CrossRef Link\)](#)
- [24] S. Tasci, and T. Güngör, "Comparison of Text Feature Selection Policies and Using an Adaptive Framework," *Expert Systems with Applications*, 40, p. 4871-4886, 2013. [Article \(CrossRef Link\)](#)
- [25] S. Gunal,S, "Hybrid Feature Selection for Text Classification," *Turkish Journal of Electrical Engineering and Computer Sciences*, 20, p. 1296-1311, 2012. [Article \(CrossRef Link\)](#)
- [26] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas., "Malware classification with recurrent networks," in *Proc. of ICASSP*, p. 1916-1920, 2015. [Article \(CrossRef Link\)](#)
- [27] T. Liu, "A Novel Text Classification Approach Based on Deep Belief Network," in *Proc. of ICONIP 2010, Part I, LNCS 6443*, p. 314-321, 2010. [Article \(CrossRef Link\)](#)

- [28] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133-3181, 2014.
- [29] R. Bermúdez-Chacón, G. H. Gonnet, K. Smith, “Automatic problem-specific hyperparameter optimization and model selection for supervised machine learning: Technical Report,” *Tech. rep.*, Zurich, 2015. [Article \(CrossRef Link\)](#)
- [30] C. Thornton, F. Hutter, H. H. Hoos, K. Leyton-Brown, “Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms,” in *KDD '13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 847-855, 2013. [Article \(CrossRef Link\)](#)
- [31] C. Tantithamthavorn, S. McIntosh, A. E. Hassan and K. Matsumoto, “An Empirical Comparison of Model Validation Techniques for Defect Prediction Models,” *IEEE Transactions on Software Engineering*, PP(99), 2016. [Article \(CrossRef Link\)](#)



Shahid Hussain received his MSc (Computer Science), MS (Software Engineering), PhD degrees from Gomal University, DIK, City University of Science and Information Technology (CUSIT), Peshawar, Pakistan, City University of Hong Kong respectively. His research interest includes the Software Design patterns and metrics, text mining, empirical studies, and software defect prediction. He has published certain papers in reputed conferences and journals.



Mohammad Rafiq Mufti received M.Sc. degree in computer science from Bahauddin Zakariya University, Multan, Pakistan and M.Sc. degree in computer engineering from Centre for Advanced Studies in Engineering (CASE) Islamabad in 1994 and 2007, respectively. He did PhD. in Electronic Engineering from Mohammad Ali Jinnah University (MAJU) Islamabad in 2012. Now he is a faculty member of Institute of Information Technology COMSATS, Vehari, Pakistan. His research interests include sliding mode control, fractional control, neural network cognitive radio network and network security.



Mohammad Khalid Sohail is currently an Assistant Professor at CIIT, Islamabad. He received his PhD from Capital University of Science & Technology (CUST), Islamabad. He had a great teaching, research, and publication experience. He has published several high impact research journal papers. His current research interest is to target the financial issues and strategies in the industries and the role of software engineering practices.



Humaira Afzal received her M. Sc. degree in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 1997. Then, she did M. Sc. in computer engineering from Centre for Advanced Studies in Engineering (CASE) Islamabad, Pakistan in 2010, and PhD. degree in computer science from School of Electrical Engineering and Computer Science University of Bradford, UK in August 2014. Currently, she is working as an assistant professor in the Institute of Computing Bahauddin Zakariya University, Multan, Pakistan. Her research interests include MAC protocol design for cognitive radio network, performance modelling, queuing theory, network security and sliding mode control.



Ghufuran Ahmed has been serving as Assistant Professor at the Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan since 2013. He received PostDoc in 2015 from Department of Computer Science and Digital Technology, Faculty of Engineering and Environment, Northumbria University, Newcastle Upon Tyne, UK. He completed Ph.D. in 2013 from Department of Computer Science, Capital University of Science and Technology, Islamabad. His area of research are Wireless Sensor Networks and Wireless Body Area Networks. He also worked as a visiting scholar at the CReWMaN Lab, Department of Computer Science and Engineering, University of Texas at Arlington in 2008–09. He is also acting as an Associate Editor for IEEE Access.



Arif Ali Khan received his BS in Software Engineering from University of Science and Technology Bannu, Pakistan in 2010. Arif has MSc by research in Information Technology from Universiti Teknologi PETRONAS, Malaysia. He has received his PhD degree from the department of computer science, City University of Hong Kong. Currently he is working as an Assistant Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. He is an active researcher in the field of empirical software engineering. His research interests include global and distributed software engineering, software process improvement, 3C's (communication, coordination, control), requirements change management and systematic reviews. He has participated in and managed several software engineering related research projects.