© 2016 IJSRSET | Volume 2 | Issue 6 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

A Framework for Collaborative Document Classification with GA-SVM

S. Chakraverty, U. Pandey, P. Dutt

[1,3]Netaji Subhas Institute of Technology Dwarka, New Delhi, Delhi, India

[2]IMS Engineering College, Ghaziabad, Ghaziabad, Uttar Pradesh, India

ABSTRACT

Text Classification has been addressed by purely statistical approaches that utilize the frequency of occurrence of significant terms as well as by tapping a range of semantic features conveyed by the text. Both approaches have proved their strengths, yet each has its own limitations when applied to corpuses with different sizes and expressive styles. This raises two interesting problems- given a corpus, how to automate the process of (i) finding an optimum blend of statistical and contextual contributions for the most appropriate classification, and (ii) determining the relative importance of different kinds of contextual features that are employed? In this paper, we address these issues by developing a Collaborative Document Classification (CDC) system that adapts according to a given corpus, the weighted contributions of statistical features, an array of lexical-semantic features derived from the WordNet ontology and categorical-semantic features obtained from the hierarchical organization of Wikipedia category pages.

Given the complexity of this multivariate problem, it is judicious to seek approximate solutions using metaheuristics. We employ a GA that embeds a multi-class SVM classifier into its fitness function evaluator to cull out an optimal mix of statistical and semantic features as tailored to a given corpus. We experimented on small as well as large data sets derived from three sources: the 20 Newsgroup corpus, the Reuters 21578 corpus and a Creative corpus that we handcrafted by collecting news articles from the Times of India news portal. Results indicate that the DC system was able to balance between statistical and context approaches and also beefed up the contributions of the most relevant semantic features for each corpus to achieve a high classification accuracy ranging from 88% to 100% with an average of 95.55%. The results highlight the significance of a collaborative DC approach that taps the power of ontological databases and can adapt to varying corpora seamlessly. The final population output by the GA contains a set of non-inferior solutions that give trade-off possibilities between recall and precision.

Keywords : Lexical semantics, WordNet ontology, Wikipedia categories, Genetic Algorithm, Multiclass SVM, Category-keyword Strength, Statistical and Contextual Document Classification.

I. INTRODUCTION

The web is a powerful storehouse of information encapsulated within innumerable documents containing unstructured as well as structured textual data. We can use these documents to sieve out valuable information pertaining to different themes or subject-wise categories. Given a set of categories, the first step is to classify each document to its most appropriate category. This function is known as Document Classification (DC). It has been applied for a wide range of applications such as e-mail management [1], sentiment analysis [2], document summarization [3], single-word question-answering [4] and web news classification [5].

The DC task can be approached either by statistical text analysis or by performing contextual analysis of the text. In [6], the authors present a survey of conventional as well as recent statistical and context based approaches that have been applied for DC. Statistical DC techniques adopt the so-called *Bag-of-Words* approach that gives importance to words based on their frequency of occurrence. Every non-trivial

term is a distinct feature whose contribution is assessed by some adaptation of the metric *Term Frequency-Inverse Document Frequency (TF-IDF)* [7]. Words that have larger number of appearances in relatively fewer documents have a greater influence on classification. We find that in all statistical approaches, the basic premise of considering every word as a potential feature generates rather high-dimensional feature spaces and raises the computational complexity of the classifier. Besides, related words that together convey some useful meaning are ignored. When a document possesses creative writing patterns with new and meaningful words, statistical techniques fail to extract substantial features despite the availability of high quality semantic content.

The above drawbacks call for a semantic analysis of text in order to extract the context presented by related parts. Context based DC has a wide scope as one can tap the power of a variety of ML techniques. Further one can utilize carefully annotated corpuses and ontology databases for semantic analysis. Substantial research has recently been directed towards context driven DC approaches [2] [8][9][10][11][12][13].

Evidently, both statistical as well as contextual approaches for DC possess their own weaknesses and strengths *vis-à-vis* different kinds of document collections. The challenge is to properly combine both these approaches to classify documents in the best possible way according to their own expressive characteristics. Some research efforts has been made in this direction by extending the frequency based method to determine the contextual score by counting joint occurrences of groups of words within a document [14] [15] [16] [17] [18].

In this paper, we address this issue by developing a Collaborative Document Classification (CDC) system that combines the strengths of both approaches by undertaking a corpus-specific adaptation of the weighted contributions of statistical and semantic classification approaches. As this involves scanning through a large multivariate solution space, it is practical to seek approximate near-optimal solutions. Genetic Algorithms (GA) are well-known population-based metasearch heuristics particularly suitable for tackling vast search spaces. We employ a Genetic Algorithm to evolve an initial population of randomized feasible solutions. Embedded into the fitness function of the GA is a Multi-class Support Vector Machine (M-SVM) that classifies the documents and calculates the classification accuracy for a given set of feature weights. The GA-SVM module performs two main functions. Firstly, it optimizes the relative contributions of statistical and context-based classification approaches. Secondly, it assigns optimal weights to the following contextual features (i) the full set of lexical-semantic features derived from the WordNet [28] and (ii) the semantically related terms and multi-word phrases extracted from Wikipedia category pages [29]. The goal is to produce the most appropriate blend of these features for achieving high quality classification.

In order to test our collaborative approach on different kinds of corpora, we chose three different sources: the 20 NewsGroups[30], the Reuters 21756 [31] corpora and we handcrafted a third dataset named "Creative Corpus" by collecting articles from the Times of India news portal [32]. Experimental results show that this approach produces solutions that yield high classification accuracy for a variety of corpuses. Moreover, the final population contains a subset of non-inferior solutions giving the benefit of trading off between precision and recall. These results highlight the significance of the proposed collaborative approach for DC.

The remaining paper is organized in the following manner. In section II, we take a tour of prior work done in the area. In section III, we present a detailed description of the proposed CDC scheme. In section IV, we reproduce experimental results and discuss their implications for DC. We conclude in section V and consider possible enhancements to our approach.

II. METHODS AND MATERIAL

1. Related Work

Seveal authors have proposed schemes to select the more significant statistical features from a large dimensional feature space [20], [23], [24], [25], [26], [27], [28]. In [24], the authors employ a hybridized Ant Colony Optimization(ACO) and GA based feature reduction technique on TF-IDF features. In [25], the authors employed K-Nearest Neighbour-GA based

feature selection scheme to select significant TF-IDF features. Suguna *et al* used rough set theory with Bee Colony Optimization for selecting the most suitable words as features for classification [26]. Alghamdi *et al*. proposed a hybrid ACO and Trace Oriented Feature Selection scheme (TOFA) for appropriate TF-IDF based feature selection and dimension reduction.

Binary and Multiclass Support Vector Machines (SVM) are sophisticated machine learning techniques that are known to generate very high accuracy classification and can handle multidimensional feature spaces [36],[37],[38]. Metasearch heuristics have also been employed to tune the Regularization parameter and kernel function parameters of SVM [24][25]. Some authors have optimized both dimensions *i.e.* the parameters of SVM as well as the features for classification [26][27]. Hidden Markov Models (HMM) and been used in conjunction with SVM for Web news classification in [5].

Recently, some research efforts have been directed towards mixed statistical and contextual DC approaches. In [14], the authors integrate the *concept score* as determined by a set of parametric context functions with the *TF-IDF* score and prove the efficacy of this approach in solving crossword puzzles. In [17], the authors parallelized both statistical and context based term weighting algorithms in order to boost overall speed-up. In [15], the authors extract groups of four to six words (*N*-grams) that repeat at least twice. These multi-word combinations represent composite features owning to their multiple instances (statistical indicator) of their co-occurrence (contextual indicator). In [16] the authors employ an Adaptive Markov model for Text Categorization called AMTC. They calculate a set of first order transition probabilities between characters and words with their *k*-predecessors, where *k* ranges from -1 to a predetermined threshold. In [18], the authors used a GA to optimize the weighted *concept standard deviation* for topic identification.

In contrast with the above works that rely only upon multi-word features within a document, we have utilized the highly structured network of lexically connected words in the WordNet ontology [28] and the well-organized categorical information compiled in Wikipedia pages [29]. These two sources are used to fulfill two purposes (i) to derive meaningful keywords for each category and (ii) to provide lexical and categorical terms that are semantically related to each token in a document. We define new metrics that quantify the semantic features of documents. Our experiments demonstrate significant improvement in our results over past approaches.

2. Framework for Collaborative DC

The proposed Collaborative DC scheme is based on supervised learning. At the outset, it populates a database of keywords for a given set of categories by extracting them automatically from the WordNet and Wikipedia ontologies. This database serves as a reference to determine the theme for each category.

Two-thirds of the documents are chosen randomly to serve as training documents and the remaining one-third form a set of testing documents. The training documents are primarily meant to train the Multi-class SVM classifier. In our scheme, they serve another purpose. The long list of keywords derived from the WordNet and Wikipedia are pruned by deleting those keywords that are not present in the training documents. This reduces the classification complexity in later stages. The accuracy of the classifier is assessed with the help of training documents.

Figure 1 illustrates the overall flow of the CDC system. There are four phases in its working, Phases A,B,C and D. The detailed working of each phase is now explained.

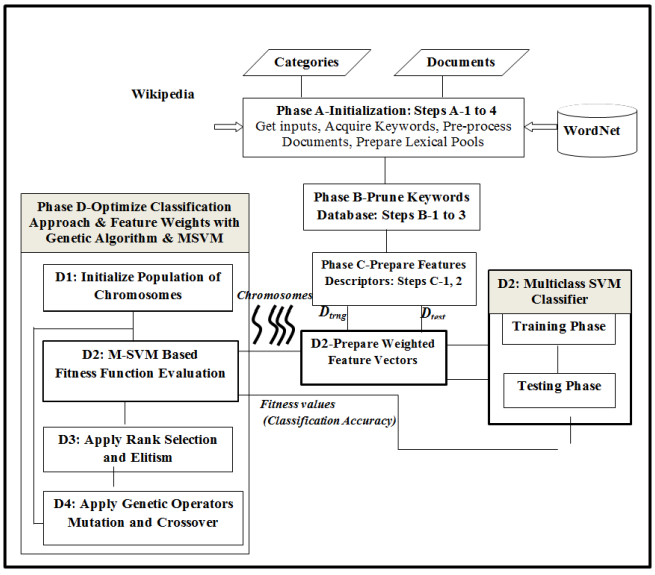

Figure 1: Organization of Collaborative Document Classification (CDC) scheme

A) **Initialization**

The initialization phase takes necessary inputs, prepares the category-keyword database, pre-processes the documents and obtains the lexically related set of words for each document term.

1) Inputs

The DC process begins by inputting the corpus whose documents $\boldsymbol{D} = \{d_1,\ldots,d_k\}$ need to be classified and the set of *m* categories $\boldsymbol{C} = \{c_1,\ldots,c_m\}$ to which they need to be assigned. Of these documents, 2/3rd are randomly selected as *training documents* $\boldsymbol{D_{trng}}$ and the remaining 1/3rd serve as *test documents* $\boldsymbol{D_{tst}}$.

2) Acquire Category Keywords

This is the initial bootstrapping step for conducting the learning process. It gathers keywords automatically for each category from two sources of hierarchically organized word concepts: the WordNet and the Wikipedia.

(i) **WordNet Keywords:** The WordNet is an online lexical inheritance database [28]. When a request is made along with a category name to the WordNet, it responds by outputting all lexical semantics for all its noun senses. These are added to the category-keyword database under their specific lexical-semantic classes. The keyword database records keywords under thirteen lexical semantic classes provided by WordNet: Synonym, Hypernym, Hyponym, Instance_Hypernym, Instance_Hyponym, Member_Meronym, Member_Holonym, Part_Meronym, Part_Holonym, Regions, Subtance_Meronym, Subtance_Holonyms and Topics. We refer to these as Lexical Keywords:

$$\boldsymbol{LK} = \{\{c_1,k_{1,1},k_{1,2}\ldots.k_{1,N(1)}.\}\ldots,\{c_j,k_{j,1},k_{j,2}\ldots..,k_{j,N(2)}\}$$

$$\ldots\{c_m,k_{m,1},k_{m,2}\ldots..,k_{m,N(m)}\}\}$$

(ii)

$\boldsymbol{LK(c)}$ refers to the WordNet keywords of the category *c*. The lexical-semantic class of a keyword *k* is denoted as *LSemC(k).*

(iii) **Keywords from Wikipedia Category pages:** The Wikipedia also serves as ontology. It categorizes information by grouping together pages on related subjects [29]. Each Wikipedia article page enlists a set of categories that are related to the topic at the bottom of the page. Each of these categories links to a category page containing a set of sub-categories and related article pages. This category page in turn, enlists its own categories thus leading to new articles and new subcategories in an iterative manner. Our DC system taps the rich categorical information provided by Wikipedia to extract keywords that have strong contextual relevance to a category but were missed by looking at WordNet alone. We adopt the following strategy to partition Wiki-keywords into different levels:

An input category name *c* is first pinged to the Wikipedia to link to the relevant article page. A collection of all category-names enlisted at the bottom of this article page and the related articles and sub-categories extracted from their respective category pages are clubbed together as *Level-1* Wiki-keywords for the input category *c*. Now each category page also enlists a new set of related categories. These in turn link to their own category pages containing more article names and sub-category names. A union of all these identifiers yields the *Level-2* Wiki-keywords. Repeating this process iteratively, one can derive keywords up to any level. For our experiments, we have used keywords till *Level-3*. Thus for any given input category *c*, the set of Wiki Keywords $\boldsymbol{WK(c)}$ are grouped into three parts, one for each of the three levels *k1,2,3* with $N_k(c)$ keywords at each level. Thus the Wiki keywords database is denoted as:

$$\boldsymbol{WK} = \{p_{c,k,l}\}: cC, k\{1,2,3\}, l\{1..N_k(c)\}$$

There are certain unique characteristics of Wikipedia category pages that distinguish it from the WordNet ontology.

- Wikipedia's category database contains single words as well as phrases. For example for the 'Computer' category, salient keywords such as *Macintosh computers*, *TRS-80 Color Computer*, *Computer architecture* and *Hardware architecture* were obtained. In order to utilize these phrases as keywords, we have incorporated functions to handle multi-words features in our DC system.
- We observed that the Wikipedia keywords for a given category rarely repeat for other categories. *Thus they are highly specific to their own categories*.

The union of the lexical keywords and Wiki keywords form the initial set of keywords for each of the categories. During next phase (Phase B) of the DC flow, the category-keyword database is pruned with the help of training documents.

3) **Preprocess Documents**

i) Stop word removal: Trivial words such as; *a, am, above* etc. do not play any role in classification. They are classed as stop words and are removed from documents. We used a reference list of 1204 stop words to remove unimportant words from all documents using the stop word list given in [33].
ii) Stemming: Stemming is the process of converting an inflected word to its base form. For stemming, we used the modified porter stemming algorithm [34].
After preprocessing, only non-trivial words in their base forms remain in the documents. These are called tokens. The set of preprocessed documents is denoted as $\boldsymbol{D}2 = \{d'_1,...,d'_k\}$. It may be noted that every document's original version *d,* as well as its preprocessed version *d*2 are preserved. As explained later, the preprocessed versions are used to sieve out unwanted WordNet keywords and the original versions are used to cull unwanted Wikipedia keywords.

4) Prepare Lexical Pools of Tokens

Each token present in the preprocessed version of a document is queried into the WordNet to extract all its lexical semantics in each of the thirteen lexical categories that were mentioned before. The full set of lexically related words that are connected to a token *w,* including itself, forms it's Lexical Pool ***LP (w).*** In this manner each original token of a preprocessed document is expanded with its Lexical Pool. This strengthens its semantic impact. We refer to this set of preprocessed documents with all tokens expanded with their own ***LPs,*** as the set of *pooled documents* $\boldsymbol{D}3 = \{d3_1,...,d3_k\}$.

B) Pruning Category Keyword Database

We observed that the total number of keywords derived from WordNet and Wikipedia together ran into thousands. For example, 5094 keywords were present in the category 'Computer' in the 20Newsgroup corpus. However, only those keywords which are present in the set of training documents would actually be used as potential features. The following steps are taken to compress the category-keyword database so that they are armed with only keywords that are required to classify a given corpus. This greatly reduces the search time when the category keyword matrix is scanned during testing.

1) **Prune Lexical Keywords:** The pseudocode in **Figure 2** describes how Lexical keywords derived from WordNet are pruned. For each category, the algorithm includes only those Lexical Keywords in the final list which occur at least once in any of the pooled training documents in $\boldsymbol{D''}$ that are labeled with that category.
2) **Prune Wikipedia Keywords:** Wikipedia's category namespace contains several multi-word phrases that may have been selected as keywords. Trivial keywords are pruned if no occurrence is found in unprocessed training documents. A process similar to that given in the pseudocode of **Figure 2** prunes Wikipedia keywords, except that (i) unprocessed training documents are utilized instead of the pooled training documents and (ii) both phrased and single keywords are searched.

Pseudocode for pruning Wordnet keywords

```
PruneWordNetKeywords(.)
Begin
 For each Category c C
    For each keyword k LK(c) {
       Set Switch OFF;
       For each pooled training document d3 □ D3
       labeled category c {
            For each token w d3 {
              If (Match(k, w)) then {
                 // Matched token found
                        Set Switch ON;
                        Break;
                 //Do not search for more tokens in
                 d3
                 }
               If (Switch is ON) Break;
                 //Do not search in more docs
            }
     If (switch is OFF) delete keyword k;
```

Pruning the database of keywords proves advantageous in the following ways:
(i) It converts a very wide set of keywords to a compact set that closely matches a given corpus.

For example the total number of keywords for the 'Computer' category was reduced from 5094 to just 164 keywords in the 20Newsgroup corpus. Keywords that may be important for a category but does not really contribute to classification due to their absence in the corpus are dropped. This reduces the time taken to prepare the test documents' raw feature vectors.

(ii) Pruning performs the process of keyword-dropping category wise. That is, a keyword may be dropped in a particular category but retained in another if needed. This boosts the *Keyword Strength* of keywords for the specific category and enhances dissimilarity among categories.

After the pruning step, the keyword database remains fixed. Hence the keywords are pre-sorted to enable a binary search.

3) **Calculate Keyword Strength:** A keyword that is presents in a single category conveys more specific information for that category than keywords that are present in several categories. The strength of a keyword k that is present one or more categories is given by:

$$IDF_w = log\frac{|D|}{|D_w|} \quad \textbf{(1)}$$

Where *presence(k,c)* is a Boolean function which returns true if keyword *k* in the given category *c*. The keyword strengths are stored for each keyword.

Keywork Strength of Lexical and Wiki Keywords: It was observed that generally, Wikipedia keywords are unique and do not repeat across categories and therefore have *KS* equal to unity. Whereas WordNet keywords sometimes repeat across categories with their *KS* value turning out to be less than unity.

C) Preparation of Document Features and Descriptors

This phase converts each document into a vector of statistical and context features.

(1) Calculate token TF_IDF: This is a statistical measure of word importance based on its frequency of occurrence. The set of preprocessed tokenized documents $\boldsymbol{D'}$ are used to generate the *TF_IDF* values of tokens. The term frequency TF_w of a token *w* is a count of the number of times it occurs in a document. Inverse document frequency $(IDF_w)$is given by

$$TF\_IDF_w = TF_w \times IDF_w \quad \textbf{(2)}$$

(2) Where $|\boldsymbol{D'}|$ is the total number of documents in a corpus and $|\boldsymbol{D'_w}|$ is the number of documents containing the word *w*. Combining these two factors, *TF_IDF* is given by their product.

$$KS_k = \frac{1}{\sum_{\forall c \in C} presence(k,c)} \quad \textbf{(3)}$$

(3) Generate Contextual Feature Descriptors

The context oriented features of all documents are extracted with the help of WordNet and Wikipedia and stored in their two respective descriptors: the Lexical-features Descriptor $\boldsymbol{T_{lex}}$ and the Wiki categorical-features Descriptor $\boldsymbol{T_{wiki}}$. We explain the processes below.

(i) **Lexical Semantic Features:** The pseudo code in **Figure 3** outlines the process of extracting features that represent lexical semantics of a document. The pooled documents $\boldsymbol{D}3$ which contain the Lexical Pools of tokens are compared with the Lexical Keywords database. Taking each of the *m* categories in turn, each token's $\boldsymbol{LP}$ is queried into the category-keyword database for a possible match with one or more of the keywords. If a match is found within any of the thirteen lexical-semantic classes within a category, then the matched keyword's Keyword Strength *KS* and lexical-semantic class of the token, *LSemC(w3)* are recorded into the Lexical features Descriptor $\boldsymbol{T_{Lex}}$ along with the identifiers for the document and the category concerned. This is illustrated in **Table 1**.

It may be emphasized that it is *not* essential for an original token to match with a keyword. Rather, each token is expanded in terms of a conceptual space (the lexical pool) where its synonyms, super-concepts, sub-concepts, components etc come into the picture and all of them are searched in the keyword database. When an original token does match, it is considered a synonym.

Pseudocode for generating lexical features of documents

LexicalKWMatch(.)

Begin
{
Initialize descriptor T_{Lex}.
For each pooled document *d*3 ***D***3
For each ***LP(w)*** *d*3
For each token *w*3 ***LP(w)***
For each Category *c* ***C***
For each Lexical Keyword *k* ***LK(c)***
If Match*(k, w*3*)* **then {**
Record in table T_{Lex} : *d*3, *c*, *w*3, keyword strength *KS(k)*,
Lexical semantic class of *w*3
}
}
End;

(i) **Wikipedia Features:** The Wikipedia keyword lists include multiword phrases that may include inflected words as well as stop words. Therefore we have to use the unprocessed versions of documents in order to extract Wikipedia based semantic features. It may be recalled that Wiki keywords for each category is partitioned into three levels based on number of links traversed to reach them.

(ii) For each of the *m* categories taken in turn (say *c)*, an unprocessed document *d* is matched with the Wiki Keywords in ***WK(c)***. The total number of matched keywords $|p^*_{c,k}|$ found in each Level *k* of ***WK*** is recorded in the Wikipedia based categorical-features Descriptor T_{wiki} as illustrated in **Table 2**. The process is repeated for every document.

Pooled Document $d3$	Category C	*Lex Pool of token w LP(w)*	Matched Tokens in *LP(w) : LP(w)) LK(c)*	Matched Keyword Strength	Lexical Semantic class *LSemC(w3)*
$d3_1$	c_1	w_1	$w3_{1,1}=w$	$KS_{1,1}$	*Synonymn*
			$w3_{1,2}$	$KS_{1,2}$	*Hypernymn*
			$w3_{1,3}$	$KS_{1,3}$	*Synonymn*
			$w3_{1,4}$	$KS_{1,4}$	*Hyponymn*
			$w3_{1,5}$	$KS_{1,5}$	*Topic*
			:	:	:
: $d3_n$	: □	: □	: □	: □	: □

Table 1: Lexical-features Descriptor T_{lex} for lexical semantic document features

Document d	Category c	Total number of tokens that matched Wiki-keywords in:		
		Level-1 $\|p^*_{c,1}\|$	Level-2 $\|p^*_{c,2}\|$	Level-3 $\|p^*_{c,3}\|$
d_1	c_1	$\|\{p^*_{1,1}\}\|$	$\|\{p^*_{1,2}\}\|$	$\|\{p^*_{1,2}\}\|$
	.....	.....	.....	.....
	c_m	$\|\{p^*_{m,1}\}\|$	$\|\{p^*_{m,1}\}\|$	$\|\{p^*_{m,1}\}\|$
: d_n	:	:	:	: .....

Table 2: Wikipedia based Categorical Feature Descriptor T_{wiki}

D) GA-SVM driven Optimization of Classification Approach and Feature Weights

The CDC system combines the collective strength of statistical features, 13 lexical semantic features and 3 levels of Wikipedia derived semantic features for effective classification. The objective is to assign appropriate weights that decide the relative contributions of these attributes towards classification. Each of these attributes can assume continuous numeric values as will be explained shortly. Thus, it is a multidimensional optimization problem with a large search space. It is therefore makes sense to adopt approximate approaches to seek solutions. Among them, GAs are well known population based evolutionary techniques well suited to tackle vast search spaces [35]. We develop a GA to explore the search space of various document features so as to yield near optimal solutions with high quality of

classification. Embedded into the fitness function evaluator of the GA is a multi-class SVM to classify the documents on the basis of weighted features and assess classification accuracy.

Chromosome structure: Figure 4 shows the structure of a chromosome. It consists of 17 genes in which the first 13 genes represent weights assigned to the respective lexical semantic class features, the next 3 genes represent weights for the total number of keyword matched tokens found in *Level-1, Level-2* and *Level-3* of the Wiki categorical keywords. The last gene decides the mix of statistical and contextual classification approach. It assigns a relative weight *H* to context based classification - and correspondingly, *(1-H)* to statistical classification.

Table 2: Wikipedia based Categorical Feature Descriptor T_{wiki}

1	2	3	4	5	6	7	8	9
W_{Syn}	W_{Hypo}	W_{Hyper}	W_{Ins_hyper}	W_{Ins_hypo}	W_{Mem_mero}	W_{Mem_holo}	W_{Part_mero}	W_{Part_holo}

Table 3 : Skeleton of the chromosome

10	11	12	13	14	15	16	17
W_{Region}	W_{Sub_mero}	W_{Sub_holo}	W_{Topic}	$W_{Level-1}$	$W_{Level-2}$	$W_{Level-3}$	H

1) A **Initialize Population:** An initial population of chromosomes is created with randomized weights between 0 and 1 assigned to each gene. The GA now launches the process of evolution from one generation to another, carrying out steps 2 to 4 described below. The process stops when the best fitness that is achieved in several generations convergences to a stable value.
2) **Calculate Chromosome Fitness:** Embedded into the fitness evaluation function is a *Multiclass Support Vector Machine (M-SVM)* classifier [36]. We now describe how the M-SVM works in concert with the GA to generate the fitness values of all chromosomes.

2.1) Prepare weighted Feature Vector: The GA selects each chromosome in turn and evaluates the feature values of each document with the help of the WordNet and Wikipedia derived feature descriptors $\boldsymbol{T_{Lex}}$ and $\boldsymbol{T_{wiki}}$ that were prepared earlier and the weights that are encoded into the selected chromosome. This generates the weighted feature vectors of all documents in the corpus that are input to M-SVM for classification. Let us consider a candidate document *d.*

a) Weighted Statistical feature values: The score for the statistical contribution TF_IDF_w of a token *w* in *d* is multiplied by the weighting factor *(1-H), H* being the weight of context based classification encoded into the chromosome.

$$S_w = (1-H) \times TF\_IDF_w \quad \textbf{(4)}$$

b) Weighted Lexical-semantic feature values: The keyword strength of each matched token in $\boldsymbol{T_{Lex}}$ (*i.e.* present in a lexical pool of d'' and also as a keyword in ***LK***) is multiplied with the weight assigned to the semantic class of the matching keyword. These values are summed up and multiplied by the context-weighting factor *H* to get the weighted lexical semantic score of *w*. The overall feature value of the token *w* in the category *c* is given by a combination of its statistical and lexical-semantic scores:

$$V_{c,w} = (1-H) \times TF\_IDF_w + H \times \sum_{w'' \in (LP(w) \cap LK(c))} KS_{w''} \times W_{LSemC(w'')} \quad \textbf{(5)}$$

Where, $V_{c,w}$ is the combined statistical and lexical score for a token w in category c, $KS_{w''}$ is the keyword strength of a matched token, LSemC(w'') is the lexical-semantic class of w'' and $W_{LSemC(w'')}$ is the

weight of this lexical-semantic class as encoded in the chromosome. The summation term is a semantic metric that quantifies the weighted contribution of a word towards lexical semantics.

c) Weighted Wiki-feature values: Each entry under the *Level-k* columns in the Wiki-categorical feature Descriptor $\boldsymbol{T_{wiki}}$ (refer Table 2) is multiplied by the corresponding wiki-level weight as given by the chromosome. This factor is normalized by dividing it with the document's word count. The resulting set of values concatenated for each category are and appended to the WordNet based lexical sematic features derived earlier.

$$U_{c,k} = \frac{\left|\{p^*_{c,k}\}\right|}{\left|N_{d'}\right|} \times W_{Level-k} \quad \textbf{(6)}$$

Where, $U_{c,k}$ is the weighted score of the wiki-categorical feature for category *c*, level *k*, $|\{p^*_{c,k}\}|$is the total number of tokens that matched with Wiki keywords at level *k* for category *c*, $|N_{d'}|$ is the total number of tokens in the pre-processed document, *i.e. d'* and W_k is the weight assigned for level *k* as encoded in the chromosome.

It may be noted that even though the unprocessed document versions are utilized to count $|\{p^*_{c,k}\}|$ in the numerator, it is normalized by the total number $|N_{d'}|$ of non-trivial tokens only in the denominator. The factor $\{p^*_{c,k}\}|/|N_{d'}|$ is a semantic metric that denotes the document's degree of matching with the *k-level Wiki-category page*s for a given input category *c*.

2.2) Conduct M-SVM training: The weighted feature vectors formed by all the tokens of all the documents generated by a chromosome are input to the M-SVM fitness evaluator. This classifier first undergoes training and learns from the *labeled feature vectors* of training documents to calculate the separating hyper-planes. M-SVM uses a Regularization Parameter *(RP)* to control the trade-off between the classifier complexity and number of non-separable points. It is used to prevent over-fitting of a model when there are large number of training documents.

2.3) Calculate fitness by M-SVM testing: The M-SVM accepts weighted feature vectors of test documents from GA and based on its acquired knowledge during training for same chromosome, predicts their categories. Next, it compares the predicted categories with the original category labels of the test documents. After all test documents have been classified, M-SVM calculates classification accuracy as:

$$Accuracy = \frac{\left|\{d : label(d) = pred(d), d \in D_{tst}\}\right|}{\left|D_{tst}\right|} \times 100 \quad \textbf{(7)}$$

Where *label(d)* is the original category labeled on a document *d* and *pred(d)* is its predicted category. GA accepts this accuracy value from M-SVM as a fitness value of the chromosome that was supplied.

3) Rank Chromosomes and apply Selection: The chromosomes are assigned ranks according to their fitness values by using the rank selection operator. The topmost tanked chromosome is passed unchanged to the next generation according to the principle of *Elitism.* This ensures a non-decreasing best fitness value along consecutive generations. Chromosomes whose fitness values fall below a threshold are rejected. The remaining chromosomes are selected to reproduce new chromosomes in the next generation with selection probabilities calculated according to their ranks [39].

4) Apply Genetic Operators: GA is a meta-heuristic which carries forward an initial randomized population of solutions through generations of evolution till optimization is reached. It scans through the search space by balancing exploration mutation and exploitation techniques to yield a set of best solutions [35].We chose single point crossover that exploits the best features of past solutions and mutation that avoids local minima by exploring new features. After selecting candidate chromosomes for generating next population by ranked selection, the single-point crossover and mutation operations are applied to generate new individuals for the next generation.

The GA re-iterates through steps 2, 3 and 4 till the best fitness converges to a stable value for a number of generations. After successful executions through required number of generations, the chromosome with the highest fitness is accepted as the desired near-optimal solution that balances best between statistical and context-based approaches and applies the most beneficial context features for high quality DC. Additionally, one can select among non-inferior solutions to choose between the quality indicators *recall* and *precision*.

III. CONCLUSION

By using this proposed model a secured path can be established for communication. The system provides security at different point in time starting from cluster head election (SLEACH), secure data transfer through session establishment CKM with inclusion of pair wise key establishment (RCD and RMCM) in case of intra-cluster communication and triple key establishment in case of inter-cluster communication and watchdog nodes with rules definition and KDD data set. Hence, as a system it provides different layer of security and monitoring. Certain rules for internal attackers have been defined in the model. The KDD dataset have been used as a protective measure in the model. The KDD dataset can be well trained and implemented in the future so that a better secured system can be implemented. Also with respect to key distribution and establishment randomized combinatorial design theory and markov chain model has been used. RMCM is surely grant security in terms of key distribution but further improvements can be made on successful key generation rate.

IV. REFERENCES

[1] S. Youn and D. McLeod, "A Comparative Study for Email Classification," in Advances and Innovations in System, Computing Science and Software Engineering, 2007, pp. 387-391.

[2] Rudy Prabowo and Mike Thelwall, "Sentiment Analysis: A Combined Approach," Intenational Journal of Informetrics, Vol. 3, Nr. 2 (2009) , p. 143-157., vol. 3, no. 2, pp. 143-157, 2009.

[3] Zengmin Geng, Jujian Zhang, Xuefei Li, Jianxia Du, and Zhengdong Liu, "Research on Web Document Summarization," in Internet Technology and Applications, 2010, pp. 1-4.

[4] Marco Ernandes, Giovanni Angelini, Marco Gori, Leonardo Rigutini, and Franco Scarselli, "An Adaptive Context-Based Algorithm for Term Weighting," in 20th international joint conference on Artificial Intelligence, 2007, pp. 2748-2753.

[5] S. Babu Rengarajan, K.G Srinivasagan Krishnalal G, "A New Text Mining Approach Based on HMM-SVM for Web News Classification," International Journal of Computer Applications, vol. 1, no. 19, pp. 98-104, 2010.

[6] Upasana Pandey and S. Chakraverty, "A Review of Text Classification Approaches for Email Management," International Journal of Engineering and Technology, vol. 3, pp. 137-144, April 2011.

[7] Yiming Yang and Jan O. Pederson, "A comparative study on Feature Selection in Text Classification," in 14th Internation Conference on Machine Learning , San Francisco, US, 1997, pp. 412-420.

[8] L. Barak, Ido Dagan, and Eyal Shnarch, "Text categorization from category name via lexical reference," in Procs of NAACL HLT, June, 2009, pp. 33-36.

[9] Stephen Bloehdorn and Andreas Hotho, "Boosting for text classification with semantic features," in MSW 2004 workshop at 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, AUG ,2004, pp. 70-87.

[10] William W. Cohen and Y. Singer, "Context Sensitive Learning Methods for Text Categorization," ACM Transaction on Information Systems, vol. Vol 17, no. 2, pp. 141-173, 1999.

[11] Dinakar Jayarajan, Dipti Deodhare, and B Ravindran, "Lexical Chains as Document Feature," in 3rd International Joint Conference on Natural Language Processing, Hyderabad, India, 2008.

[12] S. Chakraverty, Bhawna Juneja, Ashima Arora, Pratishtha Jain Upasana Pandey, "Semantic document classification using lexical chaining and fuzzy approach ," Soft Computing and Engineering, vol. 1, 2011.

[13] S. Chakraverty, Rahul Jain Upasana Pandey, "Context Driven Technique for Document Classification ," Network Security, vol. 2, no. 2, pp. 23-27, 2011.

[14] Giovanni Angelini , Marco Gori , Leonardo Rigutini , Franco Scarselli Marco Ernandes, "An Adaptive Context based algorithm for Term Weighting," in 20th International Joint Conference on Artifical intelligence, San Francisco, USA, 2007, pp. 2748-2753.

[15] Wen Zhang, Taketoshi Yoshida, and Xijin Tang, "TFIDF,LSI and Multi-word in Information Retrivel and Text Categorization," in IEEE International Conference on System, Man, Cybernetics (SMC 2008), 2008, pp. 108-113.

[16] Jin Li and Wei Yi Liu Kun Yue, "An adaptive Markov Model for Text Categorization," in 3rd International Conference on Intelligent syatem and Knoweledge Engineering, 2008, pp. 802-807.

[17] Silky Arora and Shampa Chakraverty, "A Parallel Approach to Context-based Term Weighting," in World Congress on Information and Communication Technologies, 2011, pp. 951-956.
[18] S. M. Khalessizadeh, R. Zaefarian, and S. H. Nasseri and E. Ardil, "Genetic Mining: Genetic Algorithm for topic based on concept distribution," in World Academy of Science, Engineering and Technology, 2006, pp. 144-147.
[19] Sung-Hawn Min and Ingoo Han Jumin Lee, "Hybrid Genetic Algorithm and Support Vector Machine for Backruptcy Prediction," Expert Systems with Applications, vol. 31, no. 3, pp. 652-660, 2006.
[20] Yafei Wang Wei Zhao and Dan Li, "A New Feature Selection Algorithm in Text Categorization," in International Symposium on Computer, Communication, Control and Automation, 2010, pp. 146-149.
[21] Hua Zhou and Li Zhang Xiangru Meng, "Application of Support Vector Machine and Genetic Algorithm to Network Intrusion Detection," in International Conference on Wireless Communications, Networking and Mobile Computing, 2007, pp. 2267-2269.
[22] Damian Eads et al. (2002) Genetic Algorithm and Support Vector Machins for Time Series Classification. Online]. http://users.soe.ucsc.edu/~eads/papers/eads2002.pdf
[23] Xiaoyong Liu and Hui Fu. (2012) A Hybrid Algorithm for Text Classification Problem. Online]. http://pe.ord.pl/articles/2012/1b/2.pdf
[24] Meijuan Gao and Shiru Zhou Jingwen Tian, "Research of Web Classification Mining Based on Classify Support Vector Machine," in International Colloquium on Computing, Communication, Control and Management, 2009, pp. 21-24.
[25] Fagbola Temitayo, Olbbiyisi Stephen, and Adigun Abimbola, "Hybrid GA-SVM for Efficient Features Selection in E-mail Classification," Computer Engineering and Intelligent Systems, vol. 3, no. 3, 2012.
[26] Cheng-Lung Huang and Chieh-Jen Wang, "A GA based Feature Selection and Parameters Optimization for Support Vector Machine," Expert System with Applications, vol. 31, pp. 231-240, 2006.
[27] Sheng Ding and Li Chen, "Intelligent Optimization Methods for High Dimensional Data Classification for Support Vector Machine," 2010.
[28] WordNet. Online]. http://wordnet.princeton.edu
[29] Wikipedia Categorization. Online]. http://en.wikipedia.org/wiki/Help:Category
[30] 20 NewsGroup data collection. Online]. http://people.csail.mit.edu/jrennie/20Newsgroups
[31] Reuters 21578 data collection. Online]. http://www.daviddlewis.com/resources/testcollections/reuters21578
[32] Times of India. Online]. http://timesofindia.indiatimes.com/topic
[33] Stop Word List. Online]. http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop
[34] Porter stemming. Online]. http://www.phpkode.com/scripts/item/porter-stemming-algorithm
[35] David E. Goldberg, Genetic Algorithm, 4th ed. Delhi, India: Pearson Education, 2001.
[36] Koby Crammer and Yoram Singer, "The Algorithmic Implementation of Multiclass Kernel-based Vector Machines," Journal of Machine Learning Research, vol. 2, pp. 265–292, 2001.
[37] J. Lazzaro, S. Ryckebusch, and M. A. Mahowald, "Winner-take-all networks of O(N) complexity," in Advances in neural information processing systems 1, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989.
[38] S. Sathiya Keerthi Kaibo Duan, Wei Chu, Shirish Krishnaj Shevade, and Aun Neow Poo, "Multi-Category Classification by Soft-Max Combination of Binary Classifiers," in 4th International Workshop on Multiple Classifier Systems, 2003.
[39] Noraini Mohd Razali and John Geraghty, "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP," in World Congress on Engineering, vol. II, London, July 6-8, 2011.
[40] Dino Isa, Lam Hong Lee, and V. P. Kallimani and R. Raj Kumar, "Text Document Preprocessing with the Bayes Formula for Classification using support Vector Machine," IEEE Transactions on Knoweledge and Data Engineering, vol. 20, no. 9, September 2008.
[41] Tao Peng, Fengling He, and Wenli Zuo, "Text Classification from Positive and Unlabelled Documents based on GA," in VECPAR, Brazil, p. 2006.