

**A Design of an Automatic Web Page Classification System****Abstract**

Web Page Classification is one of the major problems of the today's Internet. Many classification techniques based on the supervised Machine learning algorithms have been proposed to automatically classify Web pages. The Naive Bayesian, Support Vector Machine and K-Nearest Neighbour are the most common algorithms used to classify Web pages. In this paper, a hybrid Naive Bayesian, support vector machine, Naive Bayesian, K-Nearest Neighbour are proposed to classify Web Pages. The experimental results applied on 10,000 Web pages taken from the Reuters-21578, shows that the performance of using hybrid classifiers (Naive Bayesian, Support Vector Machine and K-Nearest Neighbour) is more effective than using each classifier alone.

---

**1. Introduction**

The World Wide Web (WWW) is the huge repository of information that has been growing exponentially over the years. There are more than 7 billion pages accessible on the World Wide Web with more than 3.424 million users Internet [1]. Hyper Text Mark-up Language (HTML) is used in design Web pages. The volume of information available on the Web is excessive, so it is infeasible to classify Web pages manually. Web pages classification is a process of assigning the Web page to one or more predefined category labels [2]. Classification is often described as a supervised learning problem in which the set of labelled data is used as the training set to train a classifier which later is applied to new data to determine their categories. It is an important ingredient as is evident from the popularity of Web directories such as Yahoo! [3], the Open Directory Project [4] and Looksmart [5].

The Web provides a regularly and dynamically changing environment, which makes it difficult to build a classification model that can fit to classify many different web pages. Many Web pages classification techniques have been proposed to automatically classify the Web. Supervised machine-learning algorithms, such as Support Vector Machines [6], Naive Bayesian Classifiers [7], K-Nearest Neighbor Classifier [8] and Decision Tree Algorithms [9], are considered the most common techniques used in Web classification. This paper proposes a hybrid Web page classifier based on the classical Naive Bayesian, K-Nearest Neighbor and Support Vector Machine. The proposed technique uses Web page title and body to extract the features used in the training process.

The paper is organized as follows: Section 2 describes related works on Web page classification. Section 3 describes HTML structure. Section 4 describes classifiers: K-Nearest Neighbor, Naive Bayesian and Support Vector Machine. Section 5 describes the proposed classification technique; Section 6 discusses the results of using the proposed classification technique and Section 7 presents concluding.

**2. Related Work**

Materna [10] examined the performance of four classification algorithms with several term representations, term clustering and feature selection methods for Web page classification. The best (highest)

predictive performance among the examined algorithms was achieved with the combination of support vector machine classifier with term frequency representation and mutual information based feature selection. Zhang et al.[11]utilized fuzzy K-nearest neighbour classifier for Web document classification. In order to represent the dataset, a TF/IDF (term frequency/increase document frequency) measure was adopted.In addition, membership grade was used to enhance predictive performance. The experimental results indicated that the fuzzy K-nearest neighbour classifier obtains better performance compared with the support vector machine and K-nearest neighbour classifier for Web document classification.Ozel[12]developed a genetic algorithm-based method for Web page classification. The proposed method uses both HTML tags and tags of terms as features for classification. Genetic algorithms were used to determine appropriate weight values for each feature. The experimental results indicated that, in the case of there being enough negative documents available in the training set, high predictive accuracy rates can be obtained for Web page classification. Zhong and Zou[13] presented an ensemble classification model that combines support vector machine classifiers, principal component analysis and independent component analysis methods for Web page classification. Principal component analysis was utilized for feature reduction and independent component analysis was utilized for feature selection.Choudhary and Raikwal[14]developed a Web page classification model that consists of naive Bayes and K-nearest neighbour. After pre-processing, the feature vector of a Web page was classified by the naive Bayes classifier and the K-nearest neighbour algorithm was utilized to measure the similarity between documents of training and test sets.Gunal[15]presented a hybrid feature selection scheme for text classification. The feature selection scheme consists of a filter and wrapper-based feature selection methods. The scheme examined the effectiveness of varying sizes of feature subsets, different dataset characteristics and classification algorithms.Aytug[16]examines the effectiveness of four popular feature selection methods, and the effectiveness of four different ensemble learning algorithms (Boosting, Bagging, Dagging and Random Subspace) is evaluated on four classification algorithms (naive Bayes, K-nearest neighbor algorithm, C4.5 algorithm and FURIA algorithm). The best average predictive performance is 88.1%, which is achieved with the use of consistency-based feature selection with AdaBoost and naive Bayes algorithms. In addition, Bagging and Random Subspace yielded better performances than base learning algorithms and correlation-based and consistency-based feature selection methods achieved higher performance compared with other configurations.

### 3. HTML structure

To extract HTMLstructure for representation of the Web page, we can choose how a term is representative of the page considering the HTML element [17] [18]. For an example, Web page represents the words of the title, words of the body and the words of meta- description, but meta- tag doesn't find in the most of Web pages.The words presented in the TITLE element are generally more representative of the page's content than words presented in the BODY element.

We tested three different text sources for Web page representation, namely:

- TITLE (T): the page's title.
- BODY (B): the content of the body tag.
- BT: Body, Title (BT) is combined between body and title content.

### 4. Classifiers

The classifier is a supervised function (machine learning tool) where a learned (target) attribute is categorical. It is used after the learning process to classify new data (Web page) by giving them the best target attributes (prediction). The target attribute can be one of k class membership.

**K-Nearest Neighbor classifier(KNN)** is based on learning by analogy [19]. The K-Nearest Neighbor algorithm is considered amongst the simplest of all machine learning algorithms. The object is classified with a majority vote of its neighbours by the object being stands for the class most common amongst its  $k$  nearest Neighbor.

$N$  dimensional numeric attributes described the training samples and each one of samples represents a point in an  $n$ -dimensional space. In this method, all of the training samples are stored in an  $n$ -dimensional pattern space [8]. When pattern space gave an unknown sample, a K-Nearest Neighbor classifier searches the pattern space for the  $k$  training samples which are nearest to the unknown sample. "Closeness" is defined in terms of Euclidean distance. The Euclidean distance between two points  $W_1 = (w_{11}, w_{12}, \dots, w_{1n})$  and  $W_2 = (w_{21}, w_{22}, \dots, w_{2n})$  [21] will be as

$$\text{dist}(W_1, W_2) = \sqrt{\sum_{i=1}^n (w_{1i} - w_{2i})^2} \quad (1)$$

The unknown sample is defined the most common class among its  $k$  nearest neighbor. When  $k=1$ , the unknown sample stands for the class of the training sample which is nearest to it in pattern space. Nearest neighbor classifiers are lazy learners and instances based in that they store all of the training samples and not build a classifier until a new sample(unknown case) needs to be classified [8].

**The Naïve Bayesian Classifier (NB)** is easy, simple and effective text classification algorithm that has been shown to perform very well in practice [7] [20] [24]. The basic method of NB is using the joint probabilities of words and categories to estimate the probabilities of the categories given a document. Thus the most researchers employ the NB method by applying Bayes rule:

$$P(c_j | d_i; \hat{\theta}) = \frac{P(c_j | \hat{\theta}) \prod_{k=1}^n P(w_k | c_j; \hat{\theta})^{N(w_k, d_j)}}{\sum_{r=1}^{|c|} P(c_r | \hat{\theta}) \prod_{k=1}^n P(w_k | c_r; \hat{\theta})^{N(w_k, d_j)}} \quad (2)$$

Where  $P(c_j | \hat{\theta})$  can be calculated by counting the frequency of each category  $c_j$  occurring in the training data.  $|c|$  is the number of categories.  $P(w_i | c_j)$  assigned to the probability that word  $w_i$  occurs in class  $c_j$  which maybe small in the training data, so the Laplace smoothing is chosen to estimate it.  $N(w_k, d_j)$  is assigned the number of occurrences of a word  $w_k$  in  $d_j$ ,  $n$  is assigned to the number of words in the training data.

**The Support vector machine (SVM)** is the supervised machine learning algorithm which can be used for either classification or regression challenges [22]. It is a good and powerful learning method introduced by Weimin et al [6]. There are two methods for SVM linear classifier and non-linear classifier [23]. SVM operates by finding the best hyper surface in the space of possible inputs. This is called a linear classifier because there are many hyper-planes that might classify the data. The hyper surface tries to split the positive examples from the negative examples of finding maximize the distance between the closest of the positive and negative examples to the hyper surface [23]. Axiomatic, this makes the classification corrects for data of testing, which is closer, but not identical to the data of training.

## 5. The Proposed technique for Web Page Classification

The steps of the proposed classification technique consist of four phases: Training phase, Classification phase, Improving phase and Testing phase. In the Training phase, a set of the Web page

documents is used to train the system. In the Classification phase, we use the Naive Bayes algorithm to classify the Web sites. In the Improving phase, we try to improve the performance of the system via combine the three (Naive Bayes, Support Vector Machine and K-Nearest Neighbor) algorithms. In the Testing phase, a set of experiments is conducted to evaluate the efficiency of the proposed technique.

The following subsections illustrate the details of each phase;

### 5.1. Training phase

To extract the features used in classification phase, a training process is done. Feature extraction means transforming arbitrary data (Web page) into features that can be taken by the classifiers. The training process requires that a set of categories has been defined, and the training Web page contents need to be somehow labelled with their respective category. The steps of the training process can be written as:

- 1- Insert the Web page.
- 2- Extract the features (words exist either in the title or the body).
- 3- Remove stop words and stemming.
- 4- Store the features.
- 5- Calculate the weight (probability) of each feature.

Stop-words refer to the words which are common like "the", "an", "are", "a", "is", "too", "also", "to", "two", "as", "off", ..., which have high frequency but these words haven't a value as an index word.

These words show high frequencies in all documents on Web, so we must remove these words at the start of indexation and then we can obtain higher speeds of calculation and fewer words needing to be indexed.

Stemming refers to the process that searches morphological variations for each keyword and transforms different keywords with the same meaning into single keyword. For example, the words "classify", "classifying", "classifiers", "classified", "classification", "classifiable" exist in documents are transformed into the word "classify".

To store the extracted features a SQL Server 2008 database table is used. Each record stores the following information:

- Category name.
- Feature name.
- Number of feature's occurrences (initialized to 0).

After all the features are extracted and filling the database table with all features, we start count the elements of the table to compute the probability for each feature. To compute the probability ( $P_f$ ) for a feature, the following formula is used:

$$P_F = \frac{S/T_s}{K_n/T_n + S/T_s} \quad (3)$$

Where  $S$  is the number of occurrences of feature  $F$  in a certain category,  $T_s$  is the total number of certain category in the training set,  $n$  is the number of occurrences of feature  $F$  in another category,  $T_n$  is the total number of another category in the training set, and  $K$  is a number that can be tuned to reduce false positives by giving a higher weight to number of occurrences of other category.

## 5.2. Classification Phase

In the classification phase a set of Web pages(which are different from the Web pages used in the training step ) are used, the main steps of this phase can be described as follows;

- 1- Parse the Web page and extract both the title and the body.
- 2- Extract features from the title and body.
- 3- For each extracted feature retrieve the corresponding probability from the database DB.
- 4- Extract the most interesting features from the list of features.
- 5- Calculate the total page probability (P) by combining the probabilities of the most interesting features.

The closer the P is for 0, the more likely the page is not to be certain category and the closer P is for 1, the more likely the page is to be certain category. In our implementation the used threshold to determine whether a Web page belongs to certain category is 0.9.

## 5.3. Improving phase

The purpose of the improving phase is increasing the accuracy rate of the used classifiers. In our implementation, Naïve Bayesian algorithm is used in the feature selection process and then the support vector machine is used in the classification process. At the end of this process, the accuracy rate of the classifier is calculated. According, some of the Web pages are classified correctly and the other are classified incorrectly. The Web pages that classified incorrectly are classified again K-Nearest Neighbor algorithm. The steps of the improving phase can be summarized as follows;

- Step1:** Insert Web Pages.
- Step2:** Apply Naïve Bayesian for feature selection.
- Step3:** Apply support vector machine and get the results.
- Step4:** Determine the Web pages that classified incorrectly.
- Step5:** Apply K-Nearest Neighbor on the Web pages obtained step 4.

## 5.4. Testing Phase

To make the testing of the proposed Web page classification technique easier, a built-in testing function is included. In our implementation, we used five categories (.Earrings, Mergers, Money, Grain and Crude Oil) to test the efficiency of the proposed technique. As can be seen in Figure 1, the testing phase for one category as an example; takes a file (F) specified by the user that contains all Web pages categories. When the classifier starts the classification, five files are created. If the classifier classifies a given Web page as earning for example, then the classifier copies it from the file (F) to the earning file. If the classifier classifies a given Web page as not earning then the classifier compare it with the rest categories. When the classifier completes the classification process, it traverses all directories to count:

True positive (**TP**): the outcome is correctly classified as positive.

True negative (**TN**): the outcome is correctly classified as negative.

False positive (**FP**): the outcome is incorrectly classified as positive when it is negative.

False negative (**FN**): the outcome is incorrectly classified as negative when it is positive.

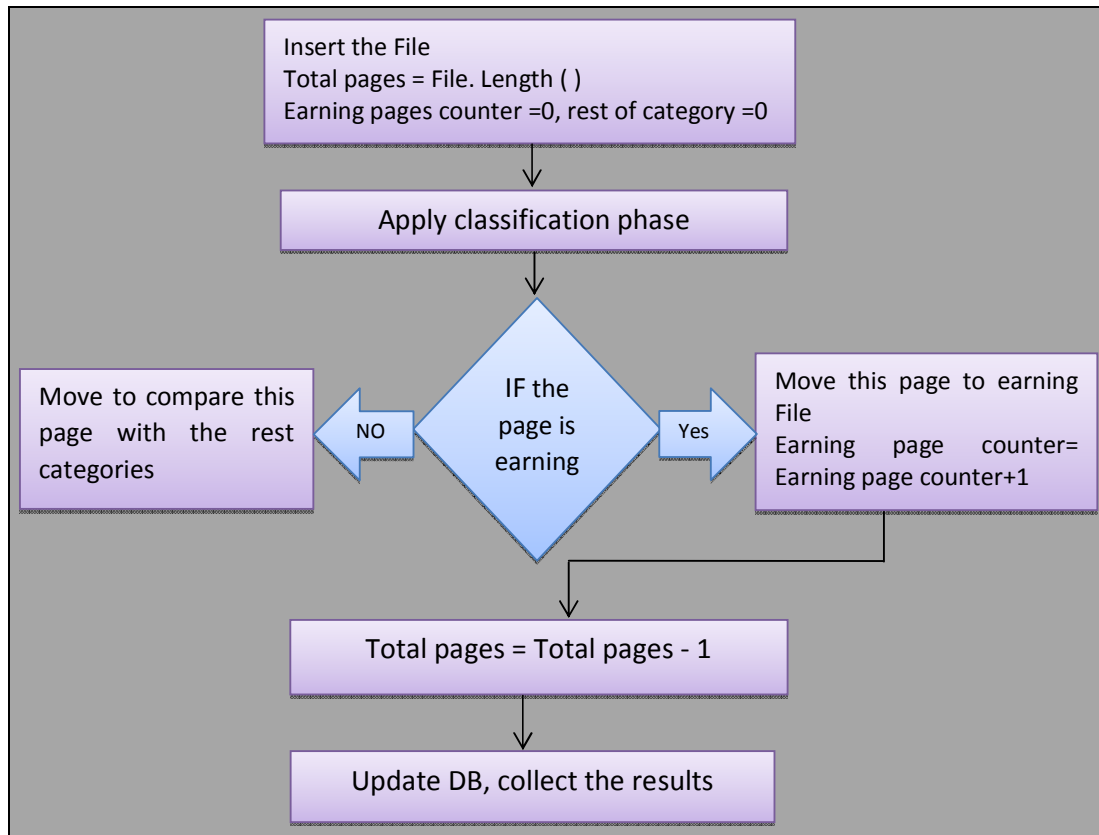


**Fig 1: Testing Phase Steps**

There are standard measures to evaluate the performance of Web classification such as precision, recall, true positive rate, false positive rate, overall accuracy and  $F_1$ -measures [26]. We calculate them after calculating the values of TP, TN, FP and FN. Precision computed as the number of correct categories assigned divided by the total number of categories assigned as equation (4) and serves as a measure of classification accuracy. Thus the higher of the precision refers the smaller the amount of false positives. Recall computed as the number of correct categories assigned divided by the total number of known correct categories as equation (5). The higher recall refers the smaller amount of missed categories. In theory, the scores of 1 are desirable for both precision and recall. In the practice, the even human assigned classifications can only achieve scores between 0.7 and 0.9; this is depending on the classification task.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{True Positive (TP) Rate} = \frac{TP}{TN + FP} \quad (6)$$

$$\text{False Positive (FP) Rate} = \frac{FP}{TN + FP} \quad (7)$$

$$\text{Overall Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

The  $F_1$ -measure is the harmonic combines or average precision and recall with equal importance into a single parameter for optimization and is defined as equation (9)

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

## 6. Experimental Results

In order to evaluate the efficiency of the classifiers discussed above, several experiments are conducted. Our experiments are done using C# programming language on an Intel Core i5, RAM 4 GB laptop running Microsoft Windows 8 and use SQL server 2008.

### 6.1. Data set

The Reuters-21578, which is considered as the standard Reuters for automatic document categorization systems [25], is used as a data set in the training and classification phases. The documents in Reuters-21578 have been collected from the Reuter's newswire in 1987. This data set consists of 21,578 Web pages which are distributed among 135 categories. We selected the categories that have largest repeated count (five category descriptions of Web pages), namely (Earrings, Mergers, Money, Grain and Crude Oil) which contain 10,000 Web pages. Randomly, 30% of these Web pages are used in the training phase and the rest (70% of these Web pages) are used in the classification and testing phases.

### 6.2. Training time

Figure 2 and Table 1 illustrate the measured execution time (second) of the training process using Naïve Bayesian, K-Nearest Neighbor and Support Vector Machine algorithms. In our implementation, the Web page title, body and both body and title (BT) are used in the training process. As can be seen in this Figure and in Table 1, the execution time of the training process using Naïve Bayesian is smaller than both k-Nearest Neighbor and Support Vector Machine. The execution time of Support Vector Machine is greater than the K-Nearest Neighbor.

**Table 1. Execution Time of training process measured by seconds**

| Text Source                    | BAYES | SVM     | KNN    |
|--------------------------------|-------|---------|--------|
| <b>TITLE</b>                   | 0.054 | 18.649  | 1.304  |
| <b>BODY</b>                    | 0.163 | 239.201 | 35.334 |
| <b>Body and Title<br/>(BT)</b> | 0.362 | 466.646 | 71.246 |

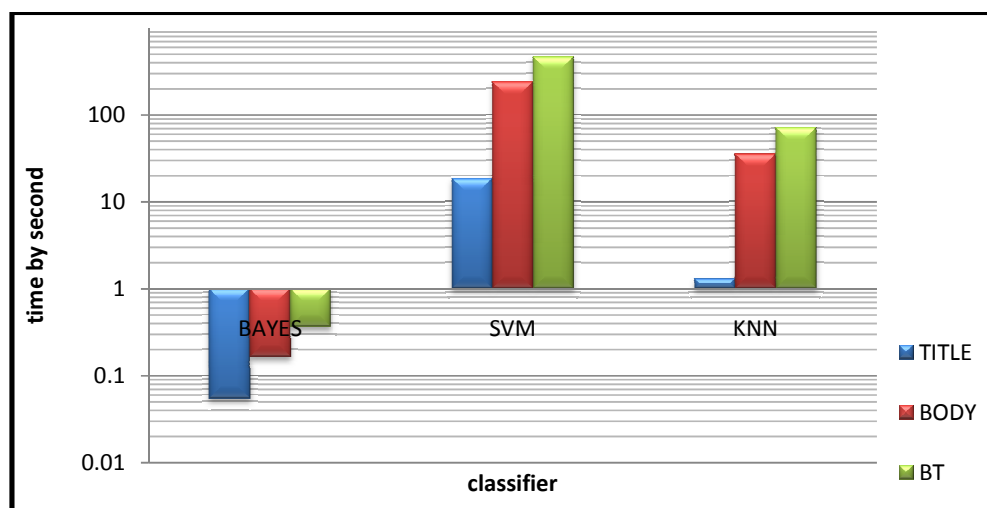


Fig 2. The execution time of the training process measured by second

### 6.3. Evaluation Experiments

#### 6.3.1. Experiment 1: Testing using Naïve Bayesian Classifier

In this section, we apply the Naive Bayesian technique (NB) using dataset selected randomly from testing data set. We have 3 test cases (title, body and, body and title(BT)). Table 2 contains the obtained true positive (TP) rate, false positive (FP) rate, precision, recall and F-measure after applying naïve Bayesian technique on each category. The Table contains also the calculated overall accuracy (using equation 8) and execution time in the case of using title, body and BT.

Table 2. Testing process using Naive Bayesian (NB)

|       |                         | TP Rate          | FP Rate      | Precision    | Recall       | F-Measure    |
|-------|-------------------------|------------------|--------------|--------------|--------------|--------------|
| TITLE | Earning                 | 0.978            | 0.192        | 0.818        | 0.978        | 0.891        |
|       | Merger                  | 0.794            | 0.022        | 0.941        | 0.794        | 0.861        |
|       | Money                   | 0.729            | 0.022        | 0.733        | 0.729        | 0.721        |
|       | Grain                   | 0.701            | 0            | 1            | 0.701        | 0.824        |
|       | Crude Oil               | 0.62             | 0.008        | 0.876        | 0.92         | 0.726        |
|       | <b>Average</b>          | <b>0.765</b>     | <b>0.049</b> | <b>0.874</b> | <b>0.765</b> | <b>0.807</b> |
|       | <b>Overall accuracy</b> | <b>0.9420317</b> |              |              |              |              |
| BODY  | Earning                 | 0.89             | 0.001        | 0.999        | 0.89         | 0.941        |
|       | Merger                  | 0.993            | 0.236        | 0.656        | 0.993        | 0.79         |
|       | Money                   | 0.535            | 0            | 0.989        | 0.535        | 0.695        |
|       | Grain                   | 0.511            | 0            | 1            | 0.511        | 0.676        |
|       | Crude Oil               | 0.403            | 0.005        | 0.869        | 0.403        | 0.551        |
|       | <b>Average</b>          | <b>0.666</b>     | <b>0.049</b> | <b>0.903</b> | <b>0.666</b> | <b>0.731</b> |
|       | <b>Overall accuracy</b> | <b>0.9327386</b> |              |              |              |              |
| BT    | Earning                 | 0.894            | 0.001        | 0.999        | 0.894        | 0.944        |
|       | Merger                  | 0.995            | 0.25         | 0.938        | 0.995        | 0.778        |
|       | Money                   | 0.5              | 0.001        | 0.987        | 0.5          | 0.664        |
|       | Grain                   | 0.446            | 0            | 1            | 0.446        | 0.617        |
|       | Crude Oil               | 0.379            | 0.003        | 0.914        | 0.379        | 0.536        |
|       | <b>Average</b>          | <b>0.643</b>     | <b>0.051</b> | <b>0.908</b> | <b>0.643</b> | <b>0.707</b> |
|       | <b>Overall accuracy</b> | <b>0.9291606</b> |              |              |              |              |

As observed in Table 2, the overall accuracy of the classification process using Web Title (94.20%) is better than the classification using Web Body (93.27%) and BT (92.92%). The average f-measure values of the classification process in the case of using Title, Body and BT are 80.7%, 73.1% and 70.7% respectively.

### 6.3.2. Experiment 2. Testing using Support Vector Machine Classifier

In this section, the support vector machine (SVM) is used to classify the considered Web page categories.

**Table 3. Testing process using support vector machine (SVM)**

|       |                         | TP Rate          | FP Rate      | Precision    | Recall       | F-Measure    |
|-------|-------------------------|------------------|--------------|--------------|--------------|--------------|
| TITLE | Earning                 | 0.976            | 0.065        | 0.929        | 0.976        | 0.952        |
|       | Merger                  | 0.94             | 0.034        | 0.924        | 0.94         | 0.932        |
|       | Money                   | 0.791            | 0.002        | 0.97         | 0.791        | 0.872        |
|       | Grain                   | 0.872            | 0.003        | 0.961        | 0.872        | 0.914        |
|       | Crude Oil               | 0.817            | 0.007        | 0.911        | 0.817        | 0.861        |
|       | <b>Average</b>          | <b>0.879</b>     | <b>0.022</b> | <b>0.939</b> | <b>0.879</b> | <b>0.906</b> |
|       | <b>Overall accuracy</b> | <b>0.9722772</b> |              |              |              |              |
| BODY  | Earning                 | 0.998            | 0.003        | 0.997        | 0.998        | 0.998        |
|       | Merger                  | 0.988            | 0.004        | 0.991        | 0.988        | 0.99         |
|       | Money                   | 1                | 0.003        | 0.961        | 1            | 0.98         |
|       | Grain                   | 0.986            | 0            | 1            | 0.986        | 0.993        |
|       | Crude Oil               | 0.965            | 0.001        | 0.988        | 0.965        | 0.976        |
|       | <b>Average</b>          | <b>0.987</b>     | <b>0.002</b> | <b>0.987</b> | <b>0.987</b> | <b>0.987</b> |
|       | <b>Overall accuracy</b> | <b>0.9967828</b> |              |              |              |              |
| BT    | Earning                 | 0.999            | 0.006        | 0.993        | 0.999        | 0.996        |
|       | Merger                  | 0.981            | 0.002        | 0.996        | 0.981        | 0.988        |
|       | Money                   | 0.994            | 0.003        | 0.966        | 0.994        | 0.98         |
|       | Grain                   | 0.986            | 0            | 0.993        | 0.986        | 0.989        |
|       | Crude Oil               | 0.972            | 0.002        | 0.972        | 0.972        | 0.972        |
|       | <b>Average</b>          | <b>0.987</b>     | <b>0.003</b> | <b>0.984</b> | <b>0.987</b> | <b>0.985</b> |
|       | <b>Overall accuracy</b> | <b>0.9961045</b> |              |              |              |              |

**Table 4. Testing process using hybrid naive Bayesian and support vector machine (NB + SVM)**

|       |                         | TP Rate          | FP Rate      | Precision    | Recall       | F-Measure    |
|-------|-------------------------|------------------|--------------|--------------|--------------|--------------|
| TITLE | Earning                 | 0.977            | 0.065        | 0.928        | 0.977        | 0.952        |
|       | Merger                  | 0.942            | 0.033        | 0.929        | 0.942        | 0.935        |
|       | Money                   | 0.775            | 0.002        | 0.97         | 0.775        | 0.862        |
|       | Grain                   | 0.867            | 0.002        | 0.959        | 0.867        | 0.911        |
|       | Crude Oil               | 0.824            | 0.008        | 0.901        | 0.824        | 0.861        |
|       | <b>Average</b>          | <b>0.877</b>     | <b>0.022</b> | <b>0.938</b> | <b>0.877</b> | <b>0.904</b> |
|       | <b>Overall accuracy</b> | <b>0.972363</b>  |              |              |              |              |
| BODY  | Earning                 | 0.998            | 0.004        | 0.995        | 0.998        | 0.997        |
|       | Merger                  | 0.988            | 0.005        | 0.99         | 0.988        | 0.989        |
|       | Money                   | 0.994            | 0.003        | 0.961        | 0.994        | 0.977        |
|       | Grain                   | 0.986            | 0            | 0.993        | 0.986        | 0.989        |
|       | Crude Oil               | 0.956            | 0            | 0.994        | 0.956        | 0.975        |
|       | <b>Average</b>          | <b>0.984</b>     | <b>0.003</b> | <b>0.987</b> | <b>0.984</b> | <b>0.985</b> |
|       | <b>Overall accuracy</b> | <b>0.996211</b>  |              |              |              |              |
| BT    | Earning                 | 0.999            | 0.006        | 0.994        | 0.999        | 0.996        |
|       | Merger                  | 0.983            | 0.001        | 0.997        | 0.983        | 0.99         |
|       | Money                   | 0.994            | 0.003        | 0.966        | 0.994        | 0.98         |
|       | Grain                   | 0.986            | 0            | 0.993        | 0.986        | 0.989        |
|       | Crude Oil               | 0.978            | 0.002        | 0.978        | 0.978        | 0.978        |
|       | <b>Average</b>          | <b>0.988</b>     | <b>0.002</b> | <b>0.985</b> | <b>0.988</b> | <b>0.987</b> |
|       | <b>Overall accuracy</b> | <b>0.9964586</b> |              |              |              |              |

As can be seen in Table 3, the overall accuracy of the considered classifier in the case of using Web Body (99.68%) is better than using Web title (97.22%) and BT (99.61%). The average f-measure values of the classification process in the case of using Title, Body and BT are 90.6%, 98.7% and 98.5% respectively.

The experimental results of using hybrid naïve Bayesian and support vector machine classifiers are given in Table 4. In our implementation, the naïve Bayesian classifier is used in the feature selection process then the support vector machine is used to classify the Web pages.

As can be seen in this Table, the overall accuracy of the considered hybrid classifier in the case of using BT (99.65%) is better than using Web Body (99.62%) and Web title (97.24%). The average f-measure of the considered hybrid classifier in the case of using title, body and BT are 90.4%, 98.5% and 98.7% respectively.

Table 2 and 4 illustrate that there is an improvement in the overall accuracy and F-measure in the case of using hybrid naïve Bayesian and support vector machine compared with using naïve Bayesian alone.

### 6.3.3. Experiment 3. Testing using K-Nearest Neighbor Classifier

In this section, the K-Nearest Neighbor is used to classify the considered Web page categories.

**Table 5. Testing process using K-Nearest Neighbor (KNN)**

|       |                         | TP Rate          | FP Rate      | Precision    | Recall       | F-Measure    |
|-------|-------------------------|------------------|--------------|--------------|--------------|--------------|
| TITLE | Earning                 | 0.983            | 0.29         | 0.806        | 0.983        | 0.885        |
|       | Merger                  | 0.684            | 0.023        | 0.928        | 0.684        | 0.787        |
|       | Money                   | 0.647            | 0.01         | 0.833        | 0.647        | 0.729        |
|       | Grain                   | 0.787            | 0.004        | 0.924        | 0.787        | 0.85         |
|       | Crude Oil               | 0                | 0            | 0            | 0            | 0            |
|       | <b>Average</b>          | <b>0.62</b>      | <b>0.065</b> | <b>0.698</b> | <b>0.62</b>  | <b>0.65</b>  |
|       | <b>Overall accuracy</b> | <b>0.9361362</b> |              |              |              |              |
| BODY  | Earning                 | 0.998            | 0.035        | 0.968        | 0.9988       | 0.983        |
|       | Merger                  | 0.957            | 0.005        | 0.989        | 0.957        | 0.973        |
|       | Money                   | 0.97             | 0.004        | 0.953        | 0.97         | 0.962        |
|       | Grain                   | 0.939            | 0.001        | 0.984        | 0.939        | 0.961        |
|       | Crude Oil               | 0                | 0            | 0            | 0            | 0            |
|       | <b>Average</b>          | <b>0.773</b>     | <b>0.009</b> | <b>0.779</b> | <b>0.773</b> | <b>0.776</b> |
|       | <b>Overall accuracy</b> | <b>0.989906</b>  |              |              |              |              |
| BT    | Earning                 | 0.996            | 0.054        | 0.952        | 0.996        | 0.974        |
|       | Merger                  | 0.931            | 0.007        | 0.985        | 0.931        | 0.957        |
|       | Money                   | 0.955            | 0.004        | 0.949        | 0.955        | 0.952        |
|       | Grain                   | 0.945            | 0.001        | 0.984        | 0.945        | 0.964        |
|       | Crude Oil               | 0                | 0            | 0            | 0            | 0            |
|       | <b>Average</b>          | <b>0.765</b>     | <b>0.013</b> | <b>0.774</b> | <b>0.765</b> | <b>0.769</b> |
|       | <b>Overall accuracy</b> | <b>0.9857434</b> |              |              |              |              |

**Table 6. Testing process using hybrid Naive Bayes and k-nearest neighbor (NB + KNN)**

|       |                         | TP Rate          | FP Rate      | Precision    | Recall       | F-Measure    |
|-------|-------------------------|------------------|--------------|--------------|--------------|--------------|
| TITLE | Earning                 | 0.994            | 0.104        | 0.903        | 0.904        | 0.946        |
|       | Merger                  | 0.904            | 0.011        | 0.974        | 0.904        | 0.937        |
|       | Money                   | 0.841            | 0.007        | 0.907        | 0.841        | 0.873        |
|       | Grain                   | 0.849            | 0            | 1            | 0.849        | 0.918        |
|       | Crude Oil               | 0.807            | 0            | 1            | 0.807        | 0.893        |
|       | <b>Average</b>          | <b>0.879</b>     | <b>0.024</b> | <b>0.957</b> | <b>0.879</b> | <b>0.914</b> |
|       | <b>Overall accuracy</b> | <b>0.973359</b>  |              |              |              |              |
| BODY  | Earning                 | 1                | 0.022        | 0.979        | 1            | 0.989        |
|       | Merger                  | 0.964            | 0.001        | 0.997        | 0.964        | 0.98         |
|       | Money                   | 0.994            | 0.005        | 0.947        | 0.994        | 0.97         |
|       | Grain                   | 0.95             | 0.001        | 0.993        | 0.95         | 0.971        |
|       | Crude Oil               | 0.96             | 0            | 1            | 0.96         | 0.98         |
|       | <b>Average</b>          | <b>0.974</b>     | <b>0.006</b> | <b>0.983</b> | <b>0.974</b> | <b>0.978</b> |
|       | <b>Overall accuracy</b> | <b>0.9933333</b> |              |              |              |              |
| BT    | Earning                 | 1                | 0.03         | 0.971        | 1            | 0.985        |
|       | Merger                  | 0.956            | 0.001        | 0.998        | 0.956        | 0.977        |
|       | Money                   | 0.982            | 0.004        | 0.959        | 0.982        | 0.97         |
|       | Grain                   | 0.961            | 0.001        | 0.992        | 0.961        | 0.976        |
|       | Crude Oil               | 0.958            | 0            | 1            | 0.958        | 0.978        |
|       | <b>Average</b>          | <b>0.971</b>     | <b>0.007</b> | <b>0.984</b> | <b>0.971</b> | <b>0.977</b> |
|       | <b>Overall accuracy</b> | <b>0.9922854</b> |              |              |              |              |

As can be seen in Table 5, the overall accuracy of the considered classifier in the case of using Web Body(98.99%) is better than using Web title (93.61%) and BT (98.57%). The average f-measure values of the classification process in the case of using Title, Body and BT are 65%%, 77.6% and 76.9% respectively.

The experimental results of using hybrid Naïve Bayesian and K-Nearest Neighbor Classifiers are given in Table 6. In our implementation, the Naïve Bayesian classifier is used to classify the Web page then the k-nearest neighbor is used to classify all Web pages that classified incorrectly again.

As can be seen in Table 6, the overall accuracy of the considered hybrid classifier in the case of using Body (99.33%) is better than using Web title (97.34%) and Web BT (99.23%). The average f-measure of the considered hybrid classifier in the case of using title, body and BT are 91.4%, 97.8% and 97.7% respectively.

Table 2, 5 and 6 illustrate that the overall accuracy and F-measure of hybrid the Naïve Bayesian classifier and K-Nearest Neighbor is better than the overall accuracy and F-measure of both the Naïve Bayesian alone and K-Nearest Neighbor alone .

#### 6.3.4. Experiment 4. Testing using hybrid Naïve Bayesian, K-Nearest Neighbor and Support Vector Machine.

In this section, the experimental results of using hybrid Naïve Bayesian, K-Nearest Neighbor and Support Vector Machine classifiers are given in Table 7.

**Table 7. Testing process using hybrid naive Bayesian, k-nearest neighbor and support vector machine (NB + KNN + SVM)**

|       |                         | TP Rate          | FP Rate      | Precision    | Recall       | F-Measure    |
|-------|-------------------------|------------------|--------------|--------------|--------------|--------------|
| TITLE | Earning                 | 0.995            | 0.048        | 0.949        | 0.995        | 0.972        |
|       | Merger                  | 0.972            | 0.006        | 0.987        | 0.972        | 0.979        |
|       | Money                   | 0.881            | 0.001        | 0.979        | 0.881        | 0.927        |
|       | Grain                   | 0.949            | 0            | 1            | 0.494        | 0.974        |
|       | Crude Oil               | 0.897            | 0            | 1            | 0.897        | 0.946        |
|       | <b>Average</b>          | <b>0.939</b>     | <b>0.011</b> | <b>0.983</b> | <b>0.939</b> | <b>0.959</b> |
|       | <b>Overall accuracy</b> | <b>0.987695</b>  |              |              |              |              |
| BODY  | Earning                 | 1                | 0            | 1            | 1            | 1            |
|       | Merger                  | 1                | 0            | 1            | 1            | 1            |
|       | Money                   | 1                | 0.002        | 0.982        | 1            | 0.991        |
|       | Grain                   | 0.986            | 0            | 1            | 0.986        | 0.993        |
|       | Crude Oil               | 0.994            | 0            | 1            | 0.994        | 0.997        |
|       | <b>Average</b>          | <b>0.996</b>     | <b>0</b>     | <b>0.996</b> | <b>0.996</b> | <b>0.996</b> |
|       | <b>Overall accuracy</b> | <b>0.9994294</b> |              |              |              |              |
| BT    | Earning                 | 1                | 0.001        | 0.999        | 1            | 1            |
|       | Merger                  | 0.998            | 0            | 1            | 0.998        | 0.999        |
|       | Money                   | 1                | 0            | 1            | 1            | 1            |
|       | Grain                   | 1                | 0            | 1            | 1            | 1            |
|       | Crude Oil               | 1                | 0            | 1            | 1            | 1            |
|       | <b>Average</b>          | <b>1</b>         | <b>0</b>     | <b>1</b>     | <b>1</b>     | <b>1</b>     |
|       | <b>Overall accuracy</b> | <b>0.9998134</b> |              |              |              |              |

As can be seen in this Table, the overall accuracy of the considered hybrid classifier in the case of using BT (both body and title) (99.98%) is better than using Web Body (99.94%) and Web title (98.78%). The average f-measure of the considered hybrid classifier in the case of using title, body and BT are 95.9%, 99.6% and 1% respectively. Comparing table 7 with the rest of the tables, it is clear that there is an improvement in the overall accuracy and F-measure from using the naïve Bayesian classifier or support vector machine classifier or k-nearest neighbor classifier.

## 7. Conclusion

In this paper, an automatic Web page classification system is introduced. The proposed system is based on the well known naïve Bayesian, support vector machine classifier and k-nearest neighbor algorithms. A hybrid Naive Bayesian + support vector machine, hybrid Naive Bayesian + K-Nearest Neighbor and hybrid Naive Bayesian + Support Vector Machine + K-Nearest Neighbor are proposed. A dataset contains 10,000 Web pages belong to 5 different categories (Earrings, Mergers, Money, Grain and Crude Oil) is used to evaluate the effectiveness of the proposed algorithms. The experimental results showed that, the performance of hybrid the Naive Bayesian + K-Nearest Neighbor + Support Vector Machine algorithms is better than Naive Bayesian alone, K-Nearest Neighbor alone and Support Vector Machine alone.

## References

- [1] <http://www.internetlivestats.com/internet-users/> last visited 1 may 2016.
- [2] Pratiksha Y. Pawar and S. H. Gawande, "A Comparative Study on Different Types of Approaches to Text Categorization", International Journal of Machine Learning and Computing, Vol. 2, No. 4, pp. 423-426, 2012.
- [3] Yahoo!, available at <http://www.yahoo.com/> last visited 1 June 2016.
- [4] Looksmart, available at <http://www.looksmart.com/> last visited 15 June 2016.
- [5] Open Directory Project, available at <http://www.dmoz.org/> last visited 1 July 2016.
- [6] Weimin. Xue, Hong. Bao, Weitong. Huan and Yuchang. Lu, "Web Page Classification Based on SVM", 6<sup>th</sup> World Congress on Intelligent Control and Automation, Dalian, China, pp. 6111-6114, 2006.
- [7] Fan Yan, Zheng C, Wang Q. Y, Cai Q. S and Liu Jo, "Web Page Classification Based on Naive Bayes Method", (In Chinese), Journal of Software, pp. 1386-1392, 2001.
- [8] Shengyi Jiang, Guansong Pang, Meiling Wu and Limin Kuang, "An improved K-nearest neighbor algorithm for text categorization" Elsevier Ltd., Vol.39, pp.1503-1509, 2011.
- [9] Pant, B., Pant, K., Pardasani, K. R, "Decision Tree Classifier for Classification of Plant and Animal Micro RNA's", Computational Intelligence and Intelligent Systems (pp.443-451). Springer Berlin Heidelberg, (2009). Retrieved from [http://link.springer.com/chapter/10.1007/978-3-642-04962-0\\_51](http://link.springer.com/chapter/10.1007/978-3-642-04962-0_51)
- [10] Materna J, "Automated web page classification", In: Proceedings of recent advances in Slavonic natural language processing, Masaryk, Czech Republic. Masaryk: University Press, pp. 84-93, December 2008.
- [11] Zhang J, Niu Y and Nie H, "Web document classification based on fuzzy k-nn algorithm", In: Proceedings of international conference on computational intelligence and security, Beijing. New York: IEEE, pp. 193-196, December 2009.
- [12] Ozel S. A., "A web page classification system based on a genetic algorithm using tagged-terms as features", Expert Systems with Applications, Vol. 28, pp. 3407-3415, 2011.
- [13] Zhong S. and Zou D., "Web page classification using ensemble of support vector machine classifiers", Journal of Networks; Vol.76, PP. 1625-1630, 2011.
- [14] Choudhary R. and Raikwal J., "An ensemble approach to enhance performance of webpage classification", International Journal of Computer Science and Information Technologies; vol.5: 5614-5619, 2014.
- [15] Gunal S, "Hybrid feature selection for text classification", Turkish Journal of Electrical Engineering and Computer Sciences; Vol.20, PP. 1296-1311, 2012.

- [16] Aytug Onan, "Classifier and feature set ensembles for web page classification", Journal of Information Science (JIS), pp.1–16, 2015.
- [17] johnwiley and sons, "HTML and CSS: Design and Build Websites" manufactured in the united states of America,2011.
- [18] The World Wide Web Consortium (W3C)"HTML 4.01 Specification" <http://www.w3.org/TR/html4/> last visited 1 June 2016.
- [19] Fang Lu QingyuanBai, "A Refined Weighted K-Nearest Neighbours Algorithm for Text Categorization", Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference ,2010.
- [20] Vidhya. K.A G.Aghila, "A Survey of Naïve Bayes Machine Learning approach in Text Document Classification", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, 2010.
- [21] Abdo Y. Alfakih, Amir Khandani, Henry Wolkowicz," Solving Euclidean distance matrix completion problems via semidefinite programming", Computational Optimiziation and Applications, Vol. 12 , pp.13–30, Part I1999.
- [22] Joachims, T. "Text categorization with support vector machines: learning with many relevant features". In Proceedings of ECML-98, 10th European Conference on Machine Learning (Chemnitz, DE), pp. 137–142, 1998.
- [23] Hanif Mohaddes Deylami, Yashwant Prasad Singh, "Cybercrime detection techniques based on support vector machines", Artificial Intelligence Research, Vol. 2, pp. 1-12, 2013.
- [24]A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification", In AAAI-98 Workshop on Learning for Text Categorization, 1998.
- [25]Lewis, D. D.: Reuters-21578 Document Corpus V1.0. <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.htm>, Last UpdatedOctober 10, 2013.
- [26]Powers, David M. W. "Evaluation: From Precision, Recall and F-Measure To Roc,Informedness, Markedness& Correlation" Journal of Machine Learning Technologies, Bioinfo Publications,Vol. 2, pp.37-63, 2011.