Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# SENTIMENT ANALYSIS USING HYBRID METHOD OF SUPPORT VECTOR MACHINE AND DECISION TREE

[1]YASSINE AL-AMRANI, [2]MOHAMED LAZAAR, [3]KAMAL EDDINE EL KADIRI
[1, 2, 3] Abdelmalek Essaâdi University, Tetuan, Morocco
E-mail : [1]alamraniyassine@gmail.com, [2]lazaarmd@gmail.com, [3]elkadiri@uae.ma

**ABSTRACT**

The exploitation of social media (forums, blogs and social networks) has become crucial due to the explosive growth of textual data from these new sources of information. Our work focuses on the sentiment analysis resulting from the messages (SMS, Facebook, Twitter...) using original techniques of search of texts. These messages can be classified as having a positive or negative feeling based on certain aspects in relation to a query based on terms. This paper presents a hybrid approach of Support Vector Machine and Decision Tree. This approach permits to ameliorate the result in terms of accuracy and CPU time.

**Keywords:** *Sentiment Analysis, Social Media, Classification, Support Vector Machine, Decision Tree.*

## 1. INTRODUCTION

Lately, new media such as social networks (Facebook, Twitter, LinkedIn ...) are developing very interesting either in terms of volume of data or according to the number of users around the world. They offer users all the possibilities to express their opinions and to exchange their ideas with the others through multiple platforms like the SMS and the emails [16].

Sentiment analysis is the part of the text mining that attempts to define the opinions, feelings and attitudes present in a text or a set of text. It is particularly used in marketing to analyse for example the comments of the Net surfers or the comparatives and tests of the bloggers or even the social networks. Sentiment analysis requires much more understanding of the language than text analysis and subject classification. Indeed, if the simplest algorithms consider only the statistics of frequency of occurrence of the words, it is usually insufficient to define the dominant opinion in a document, especially when the content is short as messages. It is the process of determining the contextual polarity of the text, that is, whether a text is positive or negative. The use of this analysis helps researchers and decision-makers better understand opinions and client satisfaction using sentiment classification techniques in order to automatically collect different perspectives on from various platforms. There has been a large amount of research in the area of sentiment classification. Traditionally most of it has focused on classifying larger pieces of text, like reviews [4].

We tried to improve the classification of the different elements of the database we used, while minimizing the execution time and maximizing the performance of the algorithms, that's why we have thought about hybridization to get a better classification and find a satisfactory result.

In this paper, a comparison of six popular classifiers was performed to classify SMS text either positive or negative (Decision Tree (DT), Support Vector Machine (SVM), Naive Bayes (NB), PART and Logistic Regression (LR)) and our approach Decision Tree Support Vector Machine (DTSVM). The rest of the paper is described as follows: Section 2 describe sentiment analysis system. Section 3 introduces applied algorithms in this field. Section 4 discusses proposed methods. Section 5 explain the results and analysis obtained. Section 6 presents the conclusion and future work.

## 2. SENTIMENT ANALYSIS SYSTEM

Sentiment analysis is the field of study that analyzes people's opinions, sentiments toward entities such as products, services, etc... [7]. A probabilistic approach for SMS classification systems has been proposed by [2]. Recently, sentiment analysis has attracted an increasing interest. It is a hard challenge for language technologies, and achieving good results is much more difficult than some people think. The task of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

automatically classifying a text written in a natural language into a positive or negative feeling, opinion or subjectivity [5], is sometimes so complicated that even different human annotators disagree on the classification to be assigned to a given text. Personal interpretation by an individual is different from others, and this is also affected by cultural factors and each person's experience. And the shorter the text, and the worse written, the more difficult the task becomes, as in the case of messages on social networks.

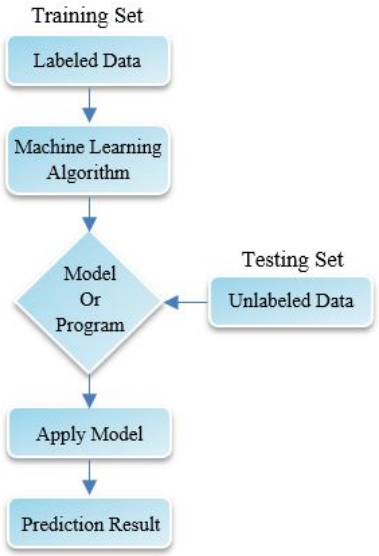



*Figure 1 Supervised Machine Learning System Approach for Sentiment Analysis*

In feature extraction, a sentence or document is broken into words to build up the feature matrix. In the matrix, each sentence or document is a row and each word form a feature as a column, and the value is the frequency count of the word in the sentence or document. Feature matrix is then passed to each classifier and their performance is evaluated [16].

In this work, we have studied the classification of sentiment using five popular machine learning algorithms, namely Decision Tree, Support Vector Machine, Naive Bayes, Logistic Regression and PART.

Decision Tree classifies data into different classes by recursively separating the feature space into two parts and assigning different classes based upon which region in the divided space a sentence is, based on its features. The Support Vector Machine method is a statistical classification approach which is based on the maximization of the margin between the instances and the separation hyper-plane. The Support Vector Machine (SVM) method was considered the best text classification method [19]. The Naïve Bayes method has been a very popular method in text categorization because its simplicity and efficiency [10]. Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). PART is a separate-and-conquer rule learner proposed by [6]. The algorithm producing sets of rules called decision lists which are ordered set of rules. A new data is compared to each rule in the list in turn, and the item is assigned the category of the first matching rule (a default is applied if no rule successfully matches) [1].

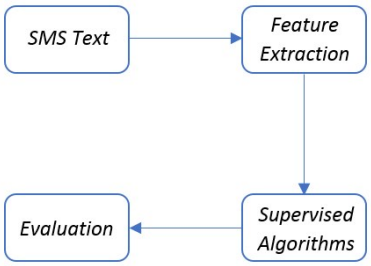



*Figure 2 Control Flow of The System*

Text classification play an important role in many applications, it assigns one or more classes to a document according to their content. Classes are selected from a previously established taxonomy (a hierarchy of categories or classes). The Text Classification API takes care of all preprocessing tasks required for automatic classification [11] [12].

This API supports a variety of text classification scenarios like:

- Binary classification like spam filtering (ham, spam) or simple sentiment analysis (positive, negative)
- Multiple class classification like selecting one category among several alternatives.

Most partitioning algorithms do not take raw text as input but numeric vectors. For this it is necessary to find a representative transformation that converts the text of the tweets to digital vectors. A family of this transformation is called Bag-of-Words (BOW).

## 3. APPLIED ALGORITHMS

This section provides a short description of all the algorithms we consider in our experimental design. All of the algorithms belong to the category

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of supervised learning methods, but we can further categorize them into neural, rule-based and statistical learning algorithms.

*Table 1: Distribution of Selected Algorithms and Their Domains*

| Group | Algorithms | Areas |
|---|---|---|
| Rules-based classifiers | PART | Classification |
| Classifiers based on decision trees | Decision Tree | |
| Bayesian networks | Naïve Bayes | |
| Classifiers 'function' | Logistic Regression | |
| | Support Vector Machine | |
| Proposed method | Decision Tree Support Vector Machine | |

### 3.1 PART

PART is a separate-and-conquer rule learner proposed by [6]. The algorithm producing sets of rules called decision lists which are ordered set of rules. A new data is compared to each rule in the list in turn, and the item is assigned the category of the first matching rule. PART builds a partial C4.5 decision tree in its each iteration and makes the best leaf into a rule. The algorithm is a combination of C4.5 and RIPPER rule learning [3].

PART (Frank, 1998) makes it possible to infer rules by the iterative generation of partial decision trees by combining two major paradigms: decision trees and the "divide and conquer" rule learning technique. It does not need to perform a global optimization to produce precise sets of rules, which brings more simplicity. It adopts the "divide and conquer" strategy because it builds a rule, removes the instances covered by this rule, and continues to create recursive rules for the remaining instances until none are left.

### 3.2 Decision Tree (DT)

A decision tree is a tree in which each node represents a choice between a number of alternative solutions, and each leaf node represents a classification or decision. The first decision tree classification algorithms are old. The two most important works were the creation of CART, by Breiman in 1984 and the creation of C4.5 by Quinlan in 1993. Decision trees are extremely intuitive and provide a graphic, speaking and easy to read representation of an individual's classification protocol. This graphical representation is in the form of a tree consisting of terminal sheets (the classes of individuals) obtained by following a path along the nodes, each node corresponding to a binary question using a variable of the dataset. It's called a decision tree because it starts with a single box (or root), which then branches off into a number of solutions, just like a tree [14].

To build a decision tree, we need to calculate Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{C} -P_i \log_2 P_i$$

S : spam or ham
C : number of classes
P : number of messages in class i

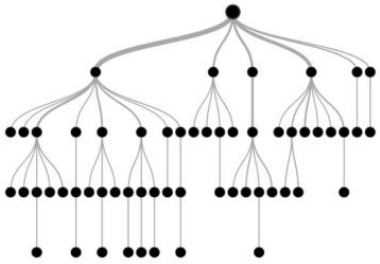


*Figure 3 Decision Tree Structure (Donaldson, 2012)*

Decision tree algorithm is an algorithm that is quick to build and test, so visualizing the tree might be important in some use cases. This can't be done in complex algorithms addressing non-linear needs such as ensemble methods.

C4.5 is the advanced version decision tree algorithm of ID3 which means the third series of 'interactive dichotomizer' procedures. For real value attributes, it is first binned into interval to form unordered nominal values. It does not consider any standard pruning procedure. C4.5 works in three main steps. First, the root node at the top node of the tree considers all samples and passes through the samples information in the second node called 'branch node'. The branch node generates rules for a group of samples based on entropy measure. In this stage, C4.5 constructs a very big tree by considering all attribute values and finalizes the decision rule by pruning. It uses a heuristic approach for pruning based on statistical significance of splits. After fixing the best rule, the branch nodes send the final target value in the last node called the 'leaf node'.

Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 3.3 Naïve Bayes (NB)

The Bayesian naive classification is a Bayesian type of simple probabilistic classification based on the Bayes theorem with a strong (or naive) independence of hypotheses. It uses a bayesian naïve classifier or bayes naïve classifier belonging to the family of linear classifiers.

The Naïve Bayes algorithm is an intuitive method; it is a simplest model that uses the probabilities of each attribute belonging to each class to make a prediction. This model works well for the categorization of the text. Naïve Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of another variable. This assumption is called class conditional independence [17]. It is classification algorithm which makes decision for unknown data set. It is based on Bayes Theorem which describe the probability of event based on its prior knowledge.

The naïve bayesian classifier provides a simple approach with clear semantics to represent, use and learn probabilistic knowledge. This method is used as part of supervised learning. The performance is to accurately predict the class of test instances.

Bayes' theorem is stated mathematically as the following equation:

$$P(H1|Ei) = \frac{P(H1) * P(Ei|H1)}{P(Ei|H1) * P(H1) + P(Ei|H2) * P(H2)}$$

Here, P(H1|Ei) is posterior probability, while P(H1) is the prior probability associated with hypothesis H1. For m different hypotheses, we have

$$P(Ei) = \sum_{j=1}^{n} P(Ei|Hj) * P(Hj)$$

Thus, we have

$$P(H1|Ei) = \frac{P(H1) * P(Ei|H1)}{P(Ei)}$$

where H1 and Ei are events and P(Ei) ≠ 0.

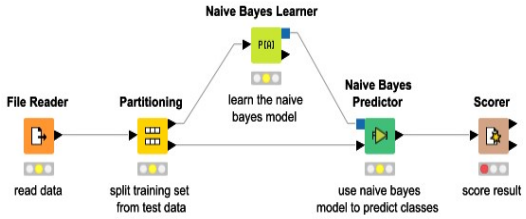



*Figure 4 Diagram Shows How Naïve Bayes Works*

### 3.4 Logistic Regression (LR)

Logistic regression is one of the most popular machine learning algorithms for binary classification. This is because it is a simple algorithm that performs very well on a wide range of problems [13]. Logistic regression corresponds to a linear regression where the dependent variable (or to explain) is binary (ie it can only take two values 0/1 or Yes / No) (Alpaydin, 2004). It is very useful for understanding or predicting the effect of one or more variables on a binary response variable.

The probability for class j with the exception of the last class is:

$$P_j(X_i) = \frac{e^{X_i B_j}}{\left(\sum_{j=1}^{k-1} e^{X_i * B_j}\right) + 1}$$

B : parameter matrix.
k : number of classes.

The last class has probability:

$$1 - \sum_{j=1}^{k-1} P_j(X_i) = \frac{1}{\left(\sum_{j=1}^{k-1} e^{(X_i * B_j)}\right) + 1}$$

### 3.5 Support Vector Machine (SVM)

This method was used for the first time in the text classification by [15]. It has proved successful in classifying opinion documents, including style (Diederich et al., 2000). The principle for support vector machine algorithm is to find a classifier, or a discrimination function, whose capacity of generalization (quality of forecasting) is the greatest possible. The examples are represented by points in a space and we look for a (hyper) plane separating at best the classes and that all the observations are the furthest from this plane [18]. The origin of the SVM algorithms is found in the methods developed in the 1960s. The principle is the separation of the learning space by a hyper plane (also called a linear surface) based on the assumption that the set d Learning consists of examples and counter examples. The SVM methods allow taking into account the problem of linearity. They first apply a mathematical transformation to the learning space using kernel functions (these functions are non-linear). Once the transformation is done, the instances can be separated linearly; it remains to find the optimal hyper plane.

Among the advantage of SVM's is that they are remarkably intolerant of the relative sizes of the number of training examples of the two classes, but the possible disadvantages of SVM's

Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7

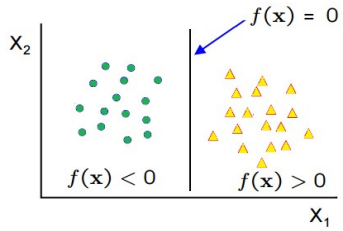
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

are that the training time can be very large if there are large numbers of training examples and execution can be slow for nonlinear SVM's.

To make our discussion of SVMs easier we will be considering a linear classifier for a binary classification problem with labels y and features x. We'll use y ∈ {−1, 1} to denote the class labels and parameters w, b: [20]

$$\mathbf{f(x) = w^T x + b}$$

w: normal to the line.
b: bias.




Choose normalization such that ($w^T$ $x_+$ + b = +1) and ($w^T$ $x_-$ + b = -1) for the positive and negative support vectors respectively.

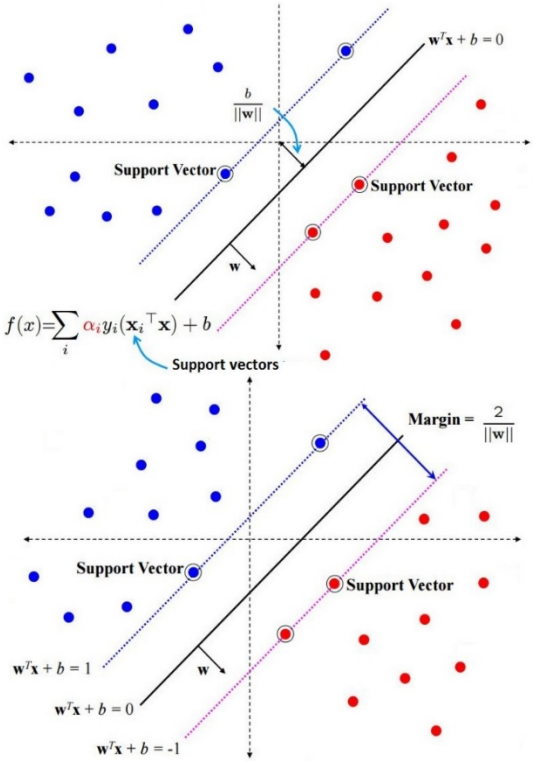



*Figure 5 Finding the Optimum Hyperplane*

Then the margin is given by:

$$\frac{w}{\|w\|} \cdot (x_+ - x_-) = \frac{w^T(x_+ - x_-)}{\|w\|}$$

$$= \frac{w^T\left(\left(\frac{+1-b}{w^T}\right) - \left(\frac{-1-b}{w^T}\right)\right)}{\|w\|} = \frac{2}{\|w\|}$$

A good choice is the hyperplane that leaves the maximum margin between the two classes, where the margin is defined as the sum of the distances of the hyperplane from the closest point of the two classes (see figure 5).

If the two classes are non-separable we can still look for the hyperplane that maximizes the margin and that minimizes a quantity proportional to the number of misclassification errors. In this case it can be shown that the solution to this problem is a linear classifier

$$f((x) = sign\left(\sum_i \alpha_i y_i \left(x_i{}^T x\right) + b\right)$$

since every coefficient corresponds to a particular data point, this means that the solution is determined by the data points associated to the non-zero coefficients. These data points, that are called support vectors, are the only ones which are relevant for the solution of the problem: all the other data points could be deleted from the data set and the same solution would be obtained. Intuitively, the support vectors are the data points that lie at the border between the two classes. Their number is usually small, and Vapnik showed that it is proportional to the generalization error of the classifier.

## 4. PROPOSED METHOD

Firstly, decision trees are known as highly efficient tools of machine learning and data mining, capable to produce accurate and easy-to-understand models. They are robust and perform well with large data in short time. On the other hand, SVM performs well on data sets that have many attributes, even if there are very few cases on which to train the model. So, we combined these two algorithms in order to obtain a very efficient algorithm with respect to time and also with the result of the classification. The advantage of DT is that it gives maximum accuracy. The low false alarm rate is the advantage of SVM.

For a classification problem with m classes, DTSVM is a binary tree with m - 1 internal nodes that each node is a binary SVM classifier and m sheet nodes. If the ranking performance is not good at the top nodes of the decision tree, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

overall performance of the classification becomes worse. Therefore, to maintain a large generalization capacity, the most separable classes should be separated at the top nodes of the decision tree.

DTSVM is applied to not only the small-scale training data but also the large-scale training data.

Suppose Di, i = 1,…,k are sets of training data included in class I, they constitute the set of active training data D

**Step 1**: calculate the separability measure in feature space $S_{ij}^H$ , i, j = 1,…,k , the $S_{ij}^H$ constitute a matrix of separability measures :

$$S^H = \begin{bmatrix} Inf & \cdots & S_{1,k}^H \\ \vdots & \ddots & \vdots \\ S_{k,1}^H & \cdots & Inf \end{bmatrix}$$

Set variable n ← k, and n indicate the number of hyperplane needs to be calculated.

**Step 2**: select the most easily separated class $i_0$ :

$$i_0 = \arg\max_{i=1,\ldots,k} S_i^H$$

$$S_i^H = \min_{\substack{j=1,\ldots,k \\ j \neq i}} S_{ij}^H$$

**Step 3**: using $D_{i0}$ and D – $D_{i0}$ as the training datasets, calculate a hyperplane $f_{i_0,\bar{i_0}}$ which separates the most easily separated class $i_0$ from the others.

Using $f_{i_0,\bar{i_0}}$ as the current root node of the decision tree.

**Step 4**: update the set of active training data D.

D ← D – $D_{i0}$,
n ← n-1;

**Step 5**: if n > 1, go to Step 2, else end.

Using cross-validation to select a classification method may yield average predictive performance substantially higher than what could be achieved with any individual method.

The default is to divide the sample into 10 folds or subsamples. For each of these 10 subsamples, a tree is built with the remaining 90% of the cases. The 10% subsample in question is then treated as a test sample, i.e. the rules from the 90% learning sample are applied to build a tree with the cases of the 10% test sample and the risk is calculated for the 10% test sample. Each of the 10% folds serve once as a test sample and serve as part of the learning sample 9 times (for a 10-fold validation).

In general, cross-validation may be viewed as a means of applying partial information about appropriate methods for classification. When we know very little about a problem, we may apply cross-validation, to select between classification strategies spanning a number of paradigms. When we know more, we may use it to select strategies within a single paradigm to select an appropriate degree of pruning in inducing a decision tree, as in the CART program (Breiman et al., 1984).

In short, cross-validation may lead to better average performance at the same time that it guards against the chance of catastrophic performance. Cross-validation and prior knowledge are best seen as complementary. Little has been done to date to help us understand how to apply them together in classification work, and this appears to be an important area for future work.

## 5. RESULTS AND DISCUSSION

In this section, DTVM will be applied to our dataset. The training and test data we used for this work were taken from the "SMS Spam Collection Data Set" which contains 5574 SMS divided into two types: positive (“ham”) and negative (“spam”).

### 5.1 Using PART Algorithm

The following table shows the result obtained using the PART algorithm:

*Table 2: Cross Validation Results for PART*

| | Spam | Ham | Total |
|---|---|---|---|
| Spam | 4766 | 61 | 4827 |
| Ham | 137 | 610 | 747 |
| Total | 4903 | 671 | 5574 |

From this table, true positives for class Spam is 4766 while false positives are 61 whereas, for class Ham, true positives are 610 and false positives is 137 i.e. diagonal elements of matrix 4766 + 610 = 5376 represents the correct messages classified and other elements 61 + 137 = 198 represents the incorrect messages.

or <= 0 AND
to <= 0 AND
2 <= 0: ham (120.0/3.0)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Â£1000 <= 0 AND
FREE <= 0 AND
call <= 0 AND
Reply <= 0 AND
Txt <= 0 AND
Your <= 0 AND
s <= 0 AND
or <= 0 AND
do <= 0 AND
you <= 0: ham (34.0)

i <= 0 AND
me <= 0 AND
Hi <= 0 AND
a > 0: spam (16.0)

i > 0: ham (6.0)

me <= 0 AND
Hi <= 0 AND
and <= 0 AND
I <= 0 AND
the > 0: spam (5.0)

and <= 0 AND
is <= 0: ham (9.0/1.0)

way <= 0: spam (8.0)

: ham (2.0)

Number of Rules: 8

This notation can be read as:

- if (("or" not in message) and ("to" not in message) and ("2" not in message)) then class(message) == ham
- if (("Â£1000" not in message) and ("FREE" not in message) and ("call" not in message)) and ("Reply" not in message)) and ("Txt" not in message)) and ("Your" not in message)) and ("s" not in message)) and ("or" not in message)) and ("do" not in message)) and ("you" not in message)) then class(message) == ham
- if (("i" not in message) and ("me" not in message) and ("Hi" not in message)) and ("a"in message) then class(message) == spam

The True Positive rate for class "Ham" is the ratio between the numbers of majority (positive) class samples which are correctly classified by the algorithm and the total numbers of majority class samples.

The True Positive rate for class "Spam" is the ratio between the numbers of minority (negative) class samples which are correctly classified by the algorithm and the total numbers of minority class samples.

- True Positive rate (T-P rate) = diagonal element / sum of relevant row.
- False Positive rate (F-P rate) = non-diagonal element / sum of relevant row.
- Precision = diagonal element / sum of relevant column.
- F-Measures (F-M) = 2 * precision * recall / (precision + recall).

For class "Spam":

- T-P rate = 4766 / (4766 + 61)= 0.987
- F-P rate = 137 / (137 + 610) = 0.183
- Precision = 4766 / (4766 + 137) = 0.972
- F-M =(2*0.972*0.987)/(0.972+0.987) = 0.979

For class "Ham":

- T-P rate = 610 / (137 + 610) = 0.816
- F-P rate = 61 / (4766 + 61) = 0.012
- Precision = 610 / (61 + 610) = 0.909
- F-M = (2*0.909*0.816)/(0.909+0.816)= 0.859

### 5.2 Using Decision Tree Algorithm

The following table shows the result obtained using Decision Tree algorithm:

*Table 3: Cross Validation Results for Decision Tree*

| | Spam | Ham | Total |
|---|---|---|---|
| Spam | 4771 | 56 | 4827 |
| Ham | 133 | 614 | 747 |
| Total | 4904 | 670 | 5574 |

From this table, true positives for class Spam is 4771 while false positives are 56 whereas, for class Ham, true positives are 614 and false positives is 133 i.e. diagonal elements of matrix 4771 + 614 = 5385 represents the correct messages classified and other elements 56 + 133 = 189 represents the incorrect messages.

Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

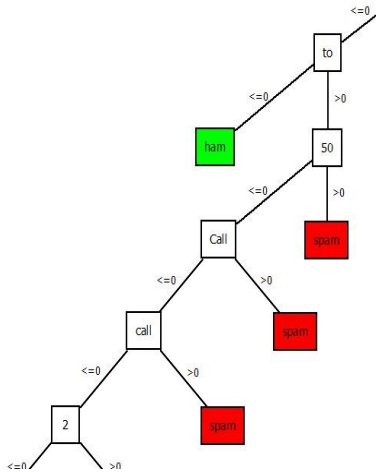



*Figure 6 Decision Tree constructed for the Train Dataset*

For class "Spam":

- T-P rate = 4771 / (4771 + 56) = 0.988
- F-P rate = 133 / (133 + 614) = 0.178
- Precision = 4771 / (4771 + 133) = 0.972
- F-M = (2*0.972*0.988)/(0.972+0.988)= 0.979

For class "Ham":

- T-P rate = 614 / (133 + 614) = 0.821
- F-P rate = 56 / (4771 + 56) = 0.011
- Precision = 614 / (56 + 614) = 0.916
- F-M = (2*0.916*0.821)/(0.916+0.821)= 0.865

### 5.3 Using Naïve Bayes Algorithm

The following table shows the result obtained using Naïve Bayes algorithm:

*Table 4: Cross Validation Results for Naïve Bayes*

| | Spam | Ham | Total |
|---|---|---|---|
| Spam | 4825 | 2 | 4827 |
| Ham | 92 | 655 | 747 |
| Total | 4917 | 657 | 5574 |

From this table, true positives for class Spam is 4825 while false positives are 2 whereas, for class Ham, true positives are 655 and false positives is 92 i.e. diagonal elements of matrix 4825 + 655 = 5480 represents the correct messages classified and other elements 2 + 92 = 94 represents the incorrect messages.

For class "Spam":

- T-P rate = 4825 / (4825 + 2) = 0.999
- F-P rate = 92 / (92 + 655) = 0.123
- Precision = 4825 / (4825 + 92) = 0.981
- F-M = (2*0.981*0.999)/(0.981+0.999)= 0.989

For class "Ham":

- T-P rate = 655 / (92 + 655) = 0.876
- F-P rate = 2 / (4825 + 2) = 4.143e-4
- Precision = 655 / (2 + 655) = 0.996
- F-M = (2*0.996*0.876)/(0.996+0.876)= 0.932

### 5.4 Using Logistic Regression Algorithm

The following table shows the result obtained using Logistic Regression algorithm:

*Table 5: Cross Validation Results for Logistic Regression*

| | Spam | Ham | Total |
|---|---|---|---|
| Spam | 4658 | 169 | 4827 |
| Ham | 63 | 684 | 747 |
| Total | 4721 | 853 | 5574 |

From this table, true positives for class Spam is 4658 while false positives are 169 whereas, for class Ham, true positives are 684 and false positives is 63 i.e. diagonal elements of matrix 4658 + 684 = 5342 represents the correct messages classified and other elements 169 + 63 = 232 represents the incorrect messages.

For class "Spam":

- T-P rate = 4658 / (4658 + 169) = 0.964
- F-P rate = 63 / (63 + 684) = 0.084
- Precision = 4658 / (4658 + 63) = 0.986
- F-M = (2*0.986*0.964)/(0.986+0.964)= 0.974

For class "Ham":

- T-P rate = 684 / (63 + 684) = 0.915
- F-P rate = 169 / (4658 + 169) = 0.035
- Precision = 684 / (169 + 684) = 0.801
- F-M = (2*0.801*0.915)/(0.801+0.915)= 0.854

### 5.5 Using Support Vector Machine Algorithm

The following table shows the result obtained using Support Vector Machine algorithm:

*Table 6: Cross Validation Results for Support Vector Machine*

| | Spam | Ham | Total |
|---|---|---|---|
| Spam | 4818 | 9 | 4827 |
| Ham | 49 | 698 | 747 |
| Total | 4867 | 707 | 5574 |

From this table, true positives for class Spam is 4818 while false positives are 9 whereas, for class Ham, true positives are 698 and false positives is 49 i.e. diagonal elements of matrix 4818 + 698 = 5516 represents the correct messages classified and other elements 9 + 49 = 58 represents the incorrect messages.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For class "Spam":

- T-P rate = 4818 / (4818 + 9) = 0.998
- F-P rate = 49 / (49 + 698) = 0.065
- Precision = 4818 / (4818 + 49) = 0.989
- F-M = (2*0.989*0.998)/(0.989+0.998)= 0.993

For class "Ham":

- T-P rate = 698 / (49 + 698) = 0.934
- F-P rate = 9 / (4818 + 9) = 0.001
- Precision = 698 / (9 + 698) = 0.987
- F-M = (2*0.987*0.934)/(0.987+0.934)= 0.959

### 5.6 Using Decision Tree Support Vector Machine Algorithm

The following table shows the result obtained using Logistic Regression algorithm:

*Table 7: Cross Validation Results for Decision Tree Support Vector Machine*

| | Spam | Ham | Total |
|---|---|---|---|
| Spam | 4827 | 0 | 4827 |
| Ham | 57 | 690 | 747 |
| Total | 4884 | 690 | 5574 |

From this table, true positives for class Spam is 4827 while false positives are 0 whereas, for class Ham, true positives are 690 and false positives is 57 i.e. diagonal elements of matrix 4827 + 690 = 5517 represents the correct messages classified and other elements 0 + 57 = 57 represents the incorrect messages.

For class "Spam":

- T-P rate = 4827 / (4827 + 0) = 1
- F-P rate = 57 / (57 + 690) = 0.076
- Precision = 4827 / (4827 + 57) = 0.988
- F-M = (2 * 0.988 * 1) / (0.988 + 1) = 0.993

For class "Ham":

- T-P rate = 690 / (57 + 690) = 0.923
- F-P rate = 0 / (4827 + 0) = 0
- Precision = 690 / (0 + 690) = 1
- F-M = (2 * 1 * 0.923) / (1 + 0.923) = 0.959

The results in terms of accuracy of the various algorithms used are presented in Table 8.

*Table 8: Number of Classified Instances*

| | C.C.M | I.C.M | A (%) | TTBM |
|---|---|---|---|---|
| PART | 5376 | 198 | 96.44 | 50.76 |
| DT | 5385 | 189 | 96.60 | 2.25 |
| NB | 5480 | 94 | 98.31 | 2.31 |
| LR | 5342 | 232 | 95.83 | 83.52 |
| SVM | 5516 | 58 | 98.95 | 7.47 |
| DTSVM | 5517 | 57 | 98.98 | 6.35 |

- C.C.M.: Correctly Classified Messages;
- I.C.M.: Incorrectly Classified Messages;
- A.: Accuracy;
- TTBM(s): Time Taken to Build Model

From Table 8 we could infer that the Decision Tree Support Vector Machine (DTSVM) perform well when compared to all other algorithms (98.98%). Improving the algorithm in different ways could improve the results further.

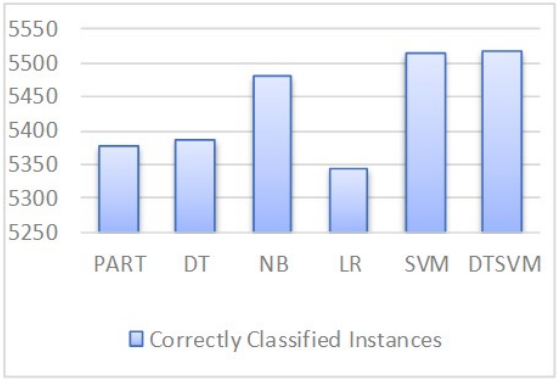



*Figure 7 Number of classified Instances for SMS*

From Figure 7 it is evident that Decision Tree Support Vector Machine (DTSVM) shows the best performance as compare to other studied algorithms.

## 6. CONCLUSION

Sentiment analysis is essential for anyone who is going to make a decision. It is helpful in different field for calculating, identifying and expressing sentiment. In this paper, we have proposed a hybrid approach of Support Vector Machine and Decision Tree. We have compared several methods with proposed approach, which are very suitable for generating rules in classification technique. From the experimental results, it is concluded that the proposed approach has high accuracy and low CPU time than the other algorithms. The reason for better results in the case of hybrid classification methodology used in this paper is since it makes use of the advantages of each of the individual traditional SVM, DT classifications methods. Although this approach has yielded interesting results, we plan to make some changes in future work to improve performance and achieve better results.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

**REFRENCES:**

[1] Aditi Mahajan, Anita Ganpati, "Performance evaluation of rule based classification algorithms", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* Vol.3, No. 10, October, 2014, pp. 3546-3550.

[2] Ahmed Ishtiaq, Donghai Guan, Tae Choong Chung, "Sms classification based on naïve bayes classifier and apriori algorithm frequent itemset", *International Journal of Machine Learning & Computing,* Vol. 4, No. 2, April 2014, pp. 183-187.

[3] Ali Shawkat, Kate A. Smith, "On learning algorithm selection for classification", *Applied Soft Computing,* Vol. 6, No. 2, January 2006, pp. 119-138.

[4] B. Pang, L. Lee, S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques", *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),* July 2002, pp. 79–86.

[5] B. Pang, L. Lee, "Opinion mining and sentiment analysis", *now publishers,* Vol. 2, No 1-2, 2008, pp. 1–135.

[6] B.R. Gaines, P. Compton, "Introduction of ripple-down rules applied to modeling large databases", *Journal of Intelligent Information Systems,* Vol. 5, No. 3, November 1995, pp. 211–228.

[7] B. Liu, "Sentiment analysis and opinion mining", *Synthesis Lectures on Human Language Technologies,* Vol.5, No. 1, April 2012, pp. 1-167.

[8] Jianlong Zhou, M. Asif Khawaja, Zhidong Li, Jinjun Sun, Yang Wang, Fang Chen, "Making machine learning useable by revealing internal states update - a transparent approach" *International Journal of Computational Science and Engineering (IJCSE),* Vol. 13, No. 4, January 2016, pp 378-389.

[9] Jun Li, Lixin Ding, Bo Li, "Differential evolution-based parameters optimisation and feature selection for support vector machine", *International Journal of Computational Science and Engineering (IJCSE),* Vol. 13, No. 4, January 2016, pp 355-363.

[10] Prem Melville, Wojciech Gryc, Richard D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification", *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,* June 28 - July 01, 2009, pp 1275-1284.

[11] M. Ettaouil, M. Lazaar, K. Elmoutaouakil, K. Haddouch, "A new algorithm for optimization of the kohonen network architectures using the continuous hopfield networks", *wseas transactions on computers,* Vol.12, No. 4, April 2013, pp 155-163.

[12] M. Ettaouil, M. Lazaar, Y. Ghanou, "Architecture optimization model for the multilayer perceptron and clustering", *Journal of Theoretical and Applied Information Technology,* Vol. 47, No. 1, January 2013, pp 064 – 072.

[13] Nádia F.F. da Silva, Eduardo R. Hruschka, Estevam R. Hruschka Jr. "Tweet sentiment analysis with classifier ensemles", *Decision Support Systems,* Vol. 66, Issue C, October 2014, pp 170-179.

[14] Rajaram Ramasamy, Appavu Balamurugan, "Suspicious e-mail detection via decision tree: a data mining approach", *Journal of computing and information technology (CIT),* Vol. 15, No. 2, 2007, pp 161-169.

[15] Thosten Joachims, "Text categorization with support vector machines: learning with many relevant features", *Proceeding of the 10th European Conference on Machine Learning,* April 21 - 23, 1998, pp 137-142.

[16] Umadevi V, "Sentiment analysis using weka", *International Journal of Engineering Trends and Technology (IJETT),* Vol. 18(4), December 2014, pp 181-183.

[17] Varsha Sahayak, Vijaya Shete, Apashabi Pathan, "Sentiment analysis on twitter data", *International Journal of Innovative Research in Advanced Engineering (IJIRAE),* Vol.2, No. 1, January 2015, pp 178-183.

[18] Witten Ian H., Eibe Frank, "Data mining: practical machine learning tools and techniques", *Morgan Kaufmann,* June 2005, pp 1-560, by Elsevier.

[19] Xia Rui, Chengqing Zong, Shoushan Li, "Ensemble of feature sets and classification algorithms for sentiment classification", *Information Sciences,* Vol.181, No. 6, 15 March 2011, pp 1138–1152.

[20] Y. Al-Amrani, M. Lazaar, K. E. Elkadiri, "Sentiment Analysis using supervised classification algorithms," Proceedings of the 2nd international Conference on Big Data, Cloud and Applications (BDCA'17). ACM, New York, NY, USA, Article 61, 8 pages. DOI: https://doi.org/10.1145/3090354.3090417.