An Ensemble Method For Spam Classification

Apurva Taunk, Srishty Bharti, Sipra Sahoo

Abstract: Spam is unsolicited and unwanted messages sent electronically. Mobile text message or SMS spam is a challenging problem due to the availability of very cheap bulk SMS packages. Due to the personal service facility and trustworthiness SMS gets higher priority. Many methods and solutions are inherited from email spam filtering as SMS spam filtering is a new arena. But it comes with its own challenges and drawbacks. Objective of this paper is to filter the mobile text message in two categories such as spam and ham. This paper aims to train the machine learning models for classifying messages into spam or ham. Along with this, an ensemble technique called as voting classifier is proposed for achieving a consistent and improved performance over the classification task. In this paper, we examined the behaviour of different machine learning algorithms which were used for the classification purpose and also created a voting classifier using those models all together as an ensemble technique with an aim of achieving a consistent and high performance over the existing classification algorithms. The features are directly extracted from the text itself as the messages have their unique characteristics which will divide them into spam or non spam classes. This paper identifies the best possible technique to work with SMS text context and has shown that using the ensemble technique gave promising results. This paper also emphasizes on the issues of data collection and availability for furthering research in this arena.

Index Terms: Classification, Evaluation measures, KNN, Machine learning, SMS, Spam, Voting Classifier

1. INTRODUCTION

Short Messaging Service (SMS) is used as an alternative for communication & transformation of information. The popularity of short message service (sms) has been growing over the last decade [1]. For businesses, these text messages are more effective than even emails. This is because while 98% of mobile users read their sms by the end of the day, about 80% of the emails remain unopened. Most of the spam messages are business related [2]. The popularity of sms has also given rise to the problem of sms spam, which refers to any irrelevant text messages delivered using mobile networks [3]. They are severely annoying to users. Most existing research that has attempted to filter sms spam has relied on manually identified features. Short messaging service (henceforth referred to as sms) is an integral part of today's world due to exploration of smart phones, i-pads and tablets. Everybody starts from businessman to network providers use sms to reach to the end user for driving their business profit. So many spam messages are coming to the mobile leading to the loss of customers as they unknowingly click the url link provided in the message. sms spam filtering issue is like email spam filtering except sms is having limited number of characters. "Catch words" can be used for giving emphasis upon the message and to attract the customers by giving a reply back number or any url (uniform resource locator) link that the customer can visit. Mobile spam messages have been a major problem in far east countries [4][5]. Along with some legal measures for the disturbances created due to spam messages technical supports are also required to prevent the spam messages leading to eradication of the abused messages. In Europe, spam message is the fashion and it is not considered as a major problem due to the cost of message is greater than email.

- *Apurva Taunk is currently pursuing bachlors degree program in computer science engineering in Siksha O Anusandhan (Deemed Tobe University) University, India, E-mail: apurvtaunk7@gmail.com*
- *Srishty Bharti is currently pursuing bachlors degree program in computer science engineering in Siksha O Anusandhan (Deemed Tobe University) University, India, E-mail: bhartisrishty7@gmail.com*
- *Sipra Sahoo is currently working as Assistant Professor in ComputerScience and Engineering in Shiksha O Anusandhan (Deemed Tobe University).University, India*

Many technical and practical methods have been applied to filter out spam messages from a bunch of messages. Many machine learning algorithms have been proposed for classifying the messages into spam or ham [6]. This paper is based upon many classification algorithms including an ensemble method to classify the text messages into spam or ham.

2. RELATED WORK:

Many algorithms and methods were proposed for the same task of classifying the SMS as a spam or not-spam. Algorithms such as Naïve Bayes and SVM were used to build a classifier that can classify the text [7][8]. An anti spam technique which is the modification of K-means algorithm and Naive Bayes algorithm has been proposed [9]. The modified K-means algorithm has given more precise results than Naïve Bayes classifier. Their proposed approach gave an accuracy of about 96%. Hybridization of modified K-means and SVM algorithm called as KSVM is proposed for spam classification. Classification time as well as misclassification are reduced using this method. Md. Refuel Islam [1] proposed a model that combined linear & non-linear SVM techniques. Where the linear SVM performed better classifying spam and not-spam emails. CHEN Xiao-li [3] proposed a method of spam filtering based on weighted SVM's. Their results show that the algorithm enhances the filtering performance effectively. The techniques such as Decision Tree Classifier, Multi-Layer perceptron and Naïve Bayes Classifier are used for classifying the messages into spam or ham [10]. The Multi-Layer perceptron model gave the best accuracy of about 99.3% though it has taken much time for training the model. Spam messages being so unwanted to the users and not so informative, to overcome the spams social networks were used by Boykin and Roy Chowdhury [11]. Gray and Haahr collaborative filtering methods to catch the spam messages from hams [12].

3. METHODOLOGY ADOPTED

This section describes the general design of workflow of the experiment. Machine learning algorithms have been used for classification. In addition to that an ensemble technique is proposed for evaluation and improvement of the used classification algorithms. Initially pre-processing is done by implementing many techniques such as lowercasing, stemming, stop word removal and bag of words model on the

gathered dataset for better quality input. After that many classifiers such as KNN, support vector classifier, logistic regression, multinomial naive bayes, stochastic gradient descent and random forest are applied to the data set we have used. Thus the data is trained using the dataset. Testing is done on the data to get results. An ensemble method is also proposed for evaluation of the classifiers and improving the performance. Finally at the last step of the experiment Confusion Matrix is obtained from the data set, and the results of the applied classifier are analyses and discussed.

3.1 Preprocessing:

Pre-processing is a step of converting the incomplete, inconsistent and unstructured real data into a well structured better quality data with which machine learning algorithms can work. Following methods are used to pre process the web data.

- **Lowercasing-** Lowercasing helps with the consistency and performance of expected output by napping all cases to lowercase form. Most of the spam messages contain information such as some links, some email addresses or some percentage or numbers. So it need to be removed and proper mapping of names should be provided. In the proposed work lowercasing normalizes the text leading to a consistent and better result. As a result more information is added to the database leading to more accurate and consistent classification.

- **Stemming:** The reduction of inflection in words and bringing it to their root form is known as stemming. The root word can be just a canonical form of the original word. Porter stemming algorithm is used for stemming in the proposed work.

- **Stop words removal:** Stop words are basically nothing but unnecessary words that doesn't add any sense and information to the message; they are used for binding the information and present it in a human understandable language. So, by removing these low informative words we can focus on the important words instead and see what the message is all about.

3.2 Bag of words model:

The algorithms used in machine learning purposes are generally based upon statistics and computations, so they need numbers to work with and cannot directly work with the raw text. To be more specific about the input, it is a vector of numbers (features). This bag of words model is a popular technique used in NLP for encoding the raw text data into numbers. This model represents the text as a description of occurrence of words within the document. It involves two things:

1. A vocabulary of words created using words from the text.
2. The measure of occurrence of words in the vocabulary.

This model uses the information gained as to whether a word from the vocabulary is present in the document or not, it only gives importance to the occurrence and not how they are arranged.

3.2.1 Tokenizing & word frequency count (creating the feature set):

Tokenization is the process of tokenizing or splitting a text into a list of tokens. Tokens here can be a sentence in a paragraph or words in a sentence. A dictionary of words is created using tokenization and Bag of words model where words are treated as keys and frequency of words is treated as the key value. In this work the k value is 1500 which is the most promising result.

3.3 Classifiers:

The proposed classifiers are discussed below.

3.3.1 KNN:

KNN is a non-parametric machine learning algorithm used for classification and regression. KNN is used to classify the K nearest matches in the training data and then using the label of the closest match to predict the class. KNN algorithm is based on feature similarity approach. KNN uses the following basic steps:

1. Calculate the distance
2. Find the closest neighbour
3. Vote for labels

KNN performs better with lower number of features than large number of features. As KNN calculates the distance between data points using the simple Euclidean Distance formula.

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (1)$$

The above formula takes 'n' number of dimensions called as features. The data point which is located at the minimum distance from the test point is assumed to belong to the same class. The above formula works the same in 'n' number of dimensions and therefore it can be used with 'n' number of features [13].

3.3.2 Support Vector Classifier (SVC):

Support vector machine is a discriminative classifier algorithm. The classifier separates the data points using hyper plane with the largest amount of margin. SVM finds an optimal hyper plane which helps in classifying new data points.SVM generates optimal hyper plane in an iterative manner that is used to minimize the error. The core idea of SVM is to find maximum marginal hyper plane (MMH) that best divides the dataset into classes [14]. A hyper plane is an n-1 dimensional subspace for an n-dimensional space. Any hyper plane can be written mathematically as follows:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \cdots + \beta_n * x_n = 0 \quad (2)$$

For a 2-D space, the hyper plane will be a line:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 = 0 \quad (3)$$

The region above the line satisfies the formula as follows:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 > 0 \quad (4)$$

The region below the line satisfies the formula as follows:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 < 0 \quad (5)$$

- **Tuning hyperparameters-**

Kernel: Different types of functions are used for the kernel in SVM algorithm such as linear, polynomial or the radial basis function (also called as RBF). For linear hyper plane, linear kernel is used and for non-linear hyper plane, the polynomial or the RBF functions are used. In this paper with this classification case, the linear kernel was the best fit. In SVM's, the hyper plane learns through the training data points provided in the training dataset. It uses linear algebra as a method for transforming the problem into some mathematical calculations that is used for creating the hyperplane used in the classification purpose in this paper. The mathematical representation for prediction function that is learned from the linear kernel trained from the training dataset is:

$$p(x) = B(0) + sum(b_i * (t, t_i)) \quad (6)$$

The above equation calculates the inner product of the input dataset for the input vector (t) with all the support vectors (t_i) present in the training dataset for the prediction of new input.
The coefficients B(0) and 'b_i' are calculated for each input vector from the training dataset using the learning algorithm.

3.3.3 Logistic Regression:

Logistic regression is the best fit when categorical values to be estimated for the dependent variable. When we have categorical values to be estimated for the dependent variable, Logistic Regression is the best fit. This is because of the fact that Linear Regression is unbounded and makes it unfit for classification purposes and this brings Logistic Regression into picture as its output value strictly ranges from 0 to +1 making it suitable for this task [15]. Logistic Regression models the probability of the default class and gives the output. Mathematical representation for the Logistic Regression function can be given as:

$$P(x) = \frac{e^{(b_0+b_1x)}}{1 + e^{(b_0+b_1x)}} \quad (7)$$

Logistic Regression seeks to learn values of b_0 & b_1 through training. A threshold value is set for predicting the class to which the input data belongs. Based upon the threshold value, the class of the input data is classified as to which class the input vector belongs. Doing some manipulation in the above equation given for Logistic Regression, we have:

$$\frac{P(x)}{1 - P(x)} = e^{(b_0+b_1x)} \quad (8)$$

Taking log on both sides-

$$log\left(\frac{P(x)}{1 - P(x)}\right) = b_0 + b_1x \quad (9)$$

(also called as logit equation)
This shows that the equation is linearly dependent on x. Thus having positive values in coefficients, we can have a clear classification of the input 'x' as the increase in 'x' will return a higher probability of the class to which 'x' belongs giving a clear classification result.

3.3.4 Multinomial Naïve Bayes:

Naïve Bayes classifier is based on bayes theorem holding an assumption of conditional independence between each pair of features. It is one from the family of probabilistic algorithms generally used for classification purposes in the field of machine learning. According to its assumptions, the algorithm learns its parameters observing each one of the features present in the feature set independently i.e. with no importance given to other features while considering one of the feature from the dataset. This assumption makes it the most proficient and helpful way of learning parameters in machine learning. Naïve Bayes classifier gives an amazing reasonable presentation during its classification tasks and the main reason behind this is the conditional hypothesis. Let's assume a term't' in some document'd' belonging to some class 'k'. Using the conditional hypothesis, it eliminates the conditional probability of a particular word provided a class as the relative frequency of the term 't' in 'd' that belongs to class 'k'. This process takes into account the frequency of occurrences of the term't' in the training set of documents from the class 'k'.

Text classification using the frequency count of a term as an event in a document is typically done using this model and this model- Multinomial Naïve bayes can be represented mathematically as-

$$p(X|C_k) = \left(\frac{(\sum_i X_i)!)}{\prod_i X_i!}\right) \prod_i {p_{ki}}^{x_i} \quad (10)$$

This equation measures the likelihood of observing the histogram x, where p_i is the probability that event 'i' occurs, C_k is the possible classes and this histogram(x) is a feature vector (x1,…, x_n) where x_i is the count of event 'i' observed at a particular instance having events represented as the occurrence of a term in the vocabulary/document/bag of words [16].

3.3.5 Stochastic Gradient Descent (SGD) Classifier:

Generally the models such as Linear SVM's or Logistic Regression models are trained for different tasks in machine learning and optimized using a method called SGD or Stochastic Gradient Descent which is also meant for classification. SGD can also be used as a classifier and in this paper, this is used to classify whether the text is a ham or a spam. A linear SVM is fiited in SGD classifier and is controlled by the loss parameter. The 'SGDClassifier' class is implemented with a first order SGD learning routine parameters are updated according to the following formula.

$$w = w - \eta\left(\left(\frac{\alpha \partial R(w)}{\partial w}\right) + \left(\frac{\partial L(w^T x_i + b_i y_i)}{\partial w}\right)\right) \quad (11)$$

In the above equation, the learning rate is represented by 'η' which controls the step size inside the parameter space, the intercept is represented by 'b' and is updated similarly but without the use of regularization. The default learning rate schedule of classification purposes is given by:

$$\eta^{(t)} = \frac{1}{\alpha(t_0 + t)} \quad (12)$$

Where 't_0' is calculated using the heuristic that is proposed by Leon Bottou [19] in such a way that the initial updates are comparable with the size of the weights as expected and the time step is represented by 't' [17].

3.3.6 Decision Tree Classifier:

Decision tree works on the basis of divide and conquer approach means partitioning of the dataset into cluster and empty regions.

The dataset is splitted into subsets which are further divided into subsets until a stopping criterion is reached. Based on largest information gain (IG) dataset is partitioned. Decision Tree model uses an optimized version of CART algorithm as a default approach where Gini index is used as a metric of evaluation for evaluation of each split in the dataset. The Gini score is calculated for all the rows and the subsets are created accordingly. This process continues until the complete tree is obtained [18]. The Gini index for a binary target variable can be expressed as:

$$Gini\ Score = 1 - \sum_{t=0}^{t=1} P_t^2 \quad (13)$$

And for categorical classification with 'k' different classes

$$Gini\ score = 1 - \sum_{t=0}^{t=k} P_t^2 \quad (14)$$

A depth limit is provided for the constructed tree for prevention of overfitting problem of Decision Tree model. However the Decision Tree classifiers are extremely fast at classifying unknown records and exclude unimportant features that reduce unwanted decisions and time.

3.3.7 Random Forest Classifier:

A random forest is a Meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset overcoming the problem of over fitting and improving the predictive accuracy. A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between the models is the key. The uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this good result is that the trees protect each other from their individual errors. Bagging or Bootstrap Aggregation – Decision trees are very sensitive to the data they are trained on. Small changes to the training set can result in significantly different tree structure. The random forest uses this as an advantage by allowing each individual tree to randomly sample from the dataset with replacement which results different decision trees. This process is called Bagging.

4. PROPOSED METHODOLOGY:

Voting Classifier:

Ensembling is a machine learning technique of combining the results of multiple models at once. The main principle used in this technique is that the combined knowledge of multiple models can work better and give a more accurate result as compared to a single model considered for the task. To build an ensemble, we simple trained all the models described above and picked up the best performing ones (in this case, we selected 5 models and they were – Decision Tree Classifier, Random Forest Classifier, Logistic Regression, Multinomial Naïve Bayes and SVC (Linear)) and then combined them to create an ensemble of models to have a simple per class voting scheme which is then used for classification our text input and classify them into classes of ham or spam.) This technique resulted in a more consistent and accurate result every time.

5. PSEUDOCODE:

1. names <- list ["DecisionTree", "RandomForest", "LogisticRegression", "NaiveBayes", "SVM"]
2. classifiers <- list [DecisionTreeClassifier(),
   RandomForestClassifier (),
   LogisticRegression (),
   MultinomialNB (),
   SVC (kernel = 'linear')]
3. models <- list (zip (names, classifiers))
4. modelvc <- SklearnClassifier(VotingClassifier(estimators = models, voting = hard))
5. modelvc.train (training dataset)
6. txt_features, labels <- zip(*testing dataset)
7. prediction <- modelvc.classify_many (txt_features)
8. confusion_matrix(labels, prediction)
9. accuracy <- nltk.classify.accuracy (modelvc, testing dataset) * 100

6. EXPERIMENTAL SETUP AND SIMULATION:

The task of SMS spam classification explained and proposed by the above methodology was performed in the system having 8GB of RAM, 1TB of Hdd storage and no Sdd storage running with a 64bit Intel i5 Processor with 2.7GHz of processing speed provided with a Windows 10 Operating system. The task was performed by the help of python (version-3.7.1) and the editor used was jupyter notebook. Due to the simple format of python language and its wide use in the field of Machine Learning, it was chosen for the task and jupyter notebook providing a good interface to work with. The above mentioned system specifications and software used helped to perform the proposed task and evaluate the same.

7. EVALUATION:

These metrics measures the overall performance of the classifiers used. The evaluation is done by measuring the percentage of spam detected and how many misclassifications are done by a particular model. The metrics used for model evaluation were –

7.1 Accuracy: Accuracy is defined as the total number of correct classifications made by the model upon the total number of classifications made altogether.

$$Accuracy = \left(\frac{TP + TN}{(TP + TN + FP + FN)}\right) * 100 \quad (15)$$

7.2 Precision: It is the fraction of correct classifications made by the classifier and which is relevant to the user also. Which is, In this case the fraction of non spam messages correctly classified as non spam upon total number of non-spam messages

$$Precision = \frac{TP}{(TP + FP)} \quad (16)$$

7.3 Recall: Recall is the fraction of correct classification relevant to the user (non-spam messages) upon total classification classified into user relevance (total messages classified as non-spam).

$$Recall = \frac{TP}{(TP + FN)} \quad (17)$$

7.4 F-Measure: It is the harmonic mean of precision and recall calculated.

$$F - Measure = 2 * (\frac{P * R}{(P + R)}) \quad (18)$$

7.5 Matthews Correlation Coefficient (MCC): MCC determines the quality of classification done by the classifier for a two class classification. It gives correct evaluation even when the size of the classes varies and the value of the coefficient varies between -1 to +1 with the coefficient value close to +1 represents finest performance.

$$MCC = \frac{((TP * TN) - (FN * FP))}{\sqrt{((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))}} \quad (19)$$

7.6 Spam Caught (SC): Spam caught value gives a representation of the model how well it can correctly classify the spam messages. It calculated as the ratio of caught spams to total number of spams in the dataset.

$$SC = \frac{TN}{(TN + FN)} \quad (20)$$

7.7 Blocked Hams (BH): Blocked Hams is a representation of how many misclassifications the model does during classification. It is the ratio of wrongly classified ham messages as spam messages to the total number of ham messages in the dataset.

$$BH = \frac{FP}{(TP + FP)} \quad (21)$$

Where TP is referred as True Positive value (i.e. amount of correct classifications of the positive instances), TN referred as True Negative value (i.e. amount of correct classifications of the negative instances.), FP referred as False Positive value (i.e. amount of misclassification of positive instances as negative instance) and FN referred as False Negative value (i.e. amount of misclassification of negative instances as positive instance.)

The accuracy comparison of all the classifiers used in the classification task here is given in the following graph:
Where the abbreviations used are KNN (K-Nearest Neighbor Classifier), DT (Decision Tree classifier), RF (Random Forest Classifier), LR (Logistic Regression Classifier), SDG (SGD Classifier), MNB (Multinomial Naïve Bayes), SVM (Support Vector Classifier (Linear)) and VC for Voting Classifier. The abbreviations are also used in the table given below which shows the performance of all the classifiers upon all the evaluation metrics. The table below describes all the evaluation metrics used for evaluating all the classifiers used and the proposed methodology used for the classification purpose.

TABLE 1

RESULTS OF ALL EVALUATION METRICS USED

Classifiers	Accuracy (%)	Precision	Recall	FM	MCC	SC	BH
KNN	96.12	0.97	0.99	0.98	0.82	0.77	0.01
Decision Tree	97.63	0.99	0.98	0.99	0.89	0.92	0.015
Random Forest	98.06	0.98	1.0	0.99	0.91	0.86	0.001
Logistic Regression	98.99	0.99	1.0	0.99	0.95	0.93	0.0008
SGD Classifier	98.2	0.99	0.99	0.99	0.92	0.91	0.008
Multinomial Naïve Bayes	98.34	0.99	0.99	0.99	0.93	0.95	0.011
SVC(linear)	98.9	0.99	1.0	0.99	0.95	0.93	0.001
Voting Classifier	98.92	0.99	1.0	0.99	0.96	0.94	0.0008

measure the performance of the voting classifier.

8. CONCLUSION:

From the above procedures of data preprocessing and methods of classifications applied on the dataset of text messages classified into spam and ham messages, we can conclude that the Random Forest Classifier, Logistic Regression, Multinomial Naïve Bayes, SVM (Linear) and the Stochastic Gradient Descent classifiers gave the best accuracy among all the classifiers applied for the classification task. Although we haven't provided any analysis about the running times of the above classifiers but we have experimentally verified that maximum of the classifiers had achieved a very high accuracy with the Logistic Regression Classifier gave the best classification accuracy of 98.99% among all the other classifiers used for the classification purpose. As the proposed model (Voting Classifier) was evaluated using all the evaluation metrics, it gave an overall high accuracy of 98.2 % along with achieving the highest MCC value of 0.96 among all other classifiers evaluated, also it has the least value of blocked hams. This resulted in a more promising approach of SMS spam classification technique giving a more consistent and more accurate result every time.

9 REFERENCES

[1] Islam, R. Md, Morshed U. Chowdhury, and Zhou Wanlei. "An innovative spam filtering model based on support vector machine." International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Vol. 2. IEEE, 2005.

[2] Yu, Bo, and Xu Zong-ben. "A comparative study for content-based dynamic spam classification using four machine learning algorithms." Knowledge-Based Systems 21.4 (2008): 355-362.

[3] Chen, Xiao-li, et al. "A method of spam filtering based on weighted support vector machines." 2009 IEEE

International Symposium on IT in Medicine & Education. Vol. 1. IEEE, 2009.

[4] Delany, Sarah Jane, Mark Buckley, and Derek Greene. "SMS spam filtering: Methods and data." Expert Systems with Applications 39.10 (2012): 9899-9908.

[5] Mathew, Kuruvilla, and Issac Biju. "Intelligent spam classification for mobile text message." Proceedings of 2011 International Conference on Computer Science and Network Technology. Vol. 1. IEEE, 2011.

[6] Renuka, D. Karthika, et al. "Spam classification based on supervised learning using machine learning techniques." 2011 International Conference on Process Automation, Control and Computing. IEEE, 2011.

[7] Metsis, Vangelis, Androutsopoulos Ion, and Paliouras Georgios. "Spam filtering with naive bayes-which naive bayes?." CEAS. Vol. 17. 2006.

[8] Caruana, Godwin, Li Maozhen, and Qi Man. "A MapReduce based parallel SVM for large scale spam filtering." 2011 eighth international conference on fuzzy systems and knowledge discovery (fskd). Vol. 4. IEEE, 2011.

[9] Tayal, K.Devendra, Amita Jain, and Kanak Meena. "Development of anti-spam technique using modified K-Means & Naive Bayes algorithm." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.

[10] Aski, Shafigh Ali, and Sourati Khalilzadeh Navid. "Proposed efficient algorithm to filter spam using machine learning techniques." Pacific Science Review A: Natural Science and Engineering 18.2 (2016): 145-149.

[11] Boykin, P. Oscar, and Vwani P. Roychowdhury. "Leveraging social networks to fight spam." Computer 4 (2005): 61-68.

[12] Gray, Alan, and Haahr Mads. "Personalised, Collaborative Spam Filtering." CEAS. 2004.

[13] Duan, Longzhen, Nan Li, and Huang Longjun. "A new spam short message classification." 2009 First International Workshop on Education Technology and Computer Science. Vol. 2. IEEE, 2009.

[14] Mccord, Michael, and M. Chuah. "Spam detection on twitter using traditional classifiers." international conference on Autonomic and trusted computing. Springer, Berlin, Heidelberg, 2011

[15] Chang, Ming-wei, Wen-tau Yih, and Christopher Meek. "Partitioned logistic regression for spam filtering." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[16] Méndez, José Ramon, et al. "A comparative impact study of attribute selection techniques on naive bayes spam filters." Industrial Conference on Data Mining. Springer, Berlin, Heidelberg, 2008.

[17] Chen, Weizhu, Gang Wang, and Zheng Chen. "Training SVMs with parallelized stochastic gradient descent." U.S. Patent No. 8,626,677. 7 Jan. 2014.

[18] Zhang, Yudong, et al. "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection." Knowledge-Based Systems 64 (2014): 22-31.

[19] L. Bottou, 2012. Stochastic gradient descent tricks. In Neural networks: Tricks of the trade (pp. 421-436). Springer, Berlin, Heidelberg.