Detection of SMS Spam Using Machine-Learning Algorithms

Fatima Zohra El Hlouli, Jamal Riffi, Mohamed Adnane Mahraz, Ali El Yahyaouy and Hamid Tairi

Abstract Short message service (SMS) is considered as one of the most popular means of communication, it allows to the mobile phone users to exchange a short text message with a low cost. Its growing popularity and its dependence on mobile phone has increased the number of attacks, caused by sending an unsolicited message like SMS spam. In this paper, we address a comparative study, between multilayer perceptron (MLP), support vector machine (SVM), random forest and k-nearest neighbors (KNN). For extracting the feature vectors the bag-of-words (BOW) and the TF-IDF methods are applied. These feature vectors are used as input for training and testing the different machine-learning classifiers mentioned above. The results of different machine-learning classifiers, based on their accuracy, precision, recall, F-measure, and ROC (receiver operating characteristic) curve have shown that the MLP outperforms SVM, random forest and KNN in SMS spam detection. Although the MLP has achieved the highest accuracy by using the BOW, than by using the TF-IDF method.

Keywords SMS spam detection · Bag-of-words · TF-IDF · MLP · SVM · Random forest and KNN

F. Z. El Hlouli (✉) · J. Riffi · M. A. Mahraz · A. El Yahyaouy · H. Tairi
Laboratory Computer, Imaging and Numerical Analysis (LIIAN), Department of Computer Science, Faculty of Sciences, University Sidi Mohammed Ben Abdellah, 30050 Atlas, Fez, Morocco
e-mail: fzelhlouli@gmail.com

J. Riffi
e-mail: riffi.jamal@gmail.com

M. A. Mahraz
e-mail: adnane_1@yahoo.fr

A. El Yahyaouy
e-mail: ali.yahyaouy@usmba.ac.ma

H. Tairi
e-mail: htairi@yahoo.fr

V. Bhateja et al. (eds.), *Embedded Systems and Artificial Intelligence*,
Advances in Intelligent Systems and Computing 1076,
https://doi.org/10.1007/978-981-15-0947-6_41

1 Introduction

SMS is a text communication platform, one SMS contains less than 160 characters. It is simple and inexpensive. It can be sent simultaneously to one or more mobiles, it no need Internet connexion like e-mail. SMS spam is generally any unwanted message sent to the mobile phone users. SMS spam is classified as scam or fraud, it can take form of text or a link to a number to call, a link to a website for more information (like card details, username, password), or a link to a website to download an application. It can include advertisement, promotions mostly on the end of the year holidays. The SMS spam is the best way to the attackers to steal the private information from the mobile phone users. So to ensure a high level of security and protection, different approaches are proposed.

Gupta et al. [1] aimed to compare eight different algorithm classifiers (SVM, NB, decision tree, logistic regression, random forest, AdaBoost, ANN, CNN) using two datasets collected from previous research [2, 3], and evaluate them basing to their accuracy, precision, recall, and CAP Curve. The experimental results show that convolutional neural network (CNN) method has achieved the highest accuracy of 99.10% and 98.25% for the two datasets.

Choudhary and Jain [4] have used ten features for classification (Presence of mathematical symbols, presence of URLs, presence of dos, presence of special symbols, presence of emotions, lowercased words, uppercased words, presence of mobile number, keyword specific, and message length), and five machine-learning algorithms namely: Naïve Bayes, logistic regression, J48, decision table and random forest. The results obtained are, that random forest classification algorithm has given a best results with 96.1% true positive rate and 1.02% false positive rate.

Uysal et al. [5] presented the impact of feature extraction on SMS spam classification combining bag-of-words model and six structural features (Message length, number of terms, uppercase character ratio, non-alphanumeric character ratio, numeric character ratio, and presence of URL), particularly for Turkish and English languages. Experimental work using Naïve Bayes and SVM classifiers indicated that the combinations of BOW and structural features has given better performance rather than bag-of-words features alone.

Popovac et al. [6] proposed CNN model composed on two convolutional layers, using TF-IDF method for extracting features, evaluated on Tiago's dataset in order to classify spam from not-spam message. The experimental results has showed that the CNN has provided better results on the same dataset with AUC score of 95.5% and accuracy of 98.4%, it has shown that CNN can be useful for SMS spam detection and similar classification.

In this paper, our aim at first is to pre-process the collection of SMS spam, and to extract the matrix of features using the BOW and the TF-IDF methods. In our work, the goal is to evaluate the performance of the different machine-learning algorithms, to provide the best algorithm for detecting SMS spam, also the best method for extracting features. The rest of this paper is organized as follows: Sect. 2 presents

methods and materials those are used in this work. Section 3 presents the results and discussions. Section 4 contains the conclusion and the future work.

2 Methods and Materials

The main objective of our comparative study is to classify the SMS messages as soon as it received on the mobile phone. In this work, after pre-processing the dataset, we extracted the features from the messages (ham and spam) using Bow and TF-IDF methods. These feature vectors are used for training and testing purposes. Figure 1 shows the system architecture of detection SMS spam using machine-learning algorithms. In the testing phase, the classifier defines whether a new message is a spam or not.

2.1 Data Pre-Processing

Our dataset is a large text file, each line corresponds to a text message. The first manipulation of cleaning the text message is to make it lowercase, removing stopwords as 'and,' 'or,' 'in' and punctuations. The tokenization is required to transform a text message to a list of words (tokens), also stemmatization is necessary for converting the different forms for the same word in their root (goes, going, gone => go).

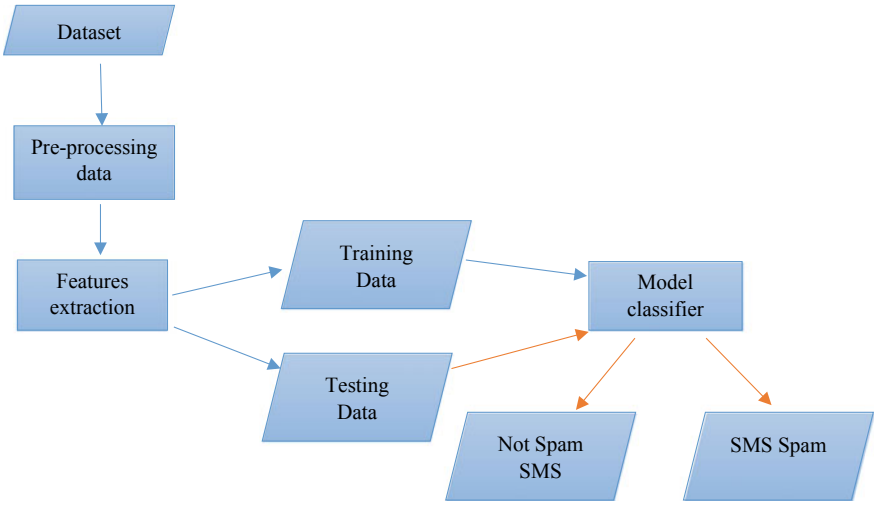

Fig. 1 Architecture of detection SMS spam using ML algorithms

```
def bag_of_words(tab_SMS):
  Initialize dic=[]
    for token in tab_SMS.tokenize():
      if(token not in dic):
       dic=dic+token
  numrows = len(tab_SMS)
  numcols = len(dic)
  bow_features = initialize the matrix with (numrows, numcols)
  for token in tab_SMS.tokenize():
    for word in dic :
      if(word in token):
        idx =getindex from word in token
        add the length of idx to bow_features in a row i
      else:
        add 0 to bow_features in a row i
    i=i+1
return bow_features
```

Fig. 2 Bag-of-words pseudo code

2.2 Features Extraction

After pre-processing the dataset, each SMS must be converted to a vector of features which the classifier can train with it, for this, bag-of-words and TF-IDF methods have been applied to transform each SMS of dataset to a vector of features. Before applying either of these methods, the first column of the dataset was changed. Ham and spam labels were converted to values 0 and 1. The two methods mentioned above are described as:

2.2.1 Bag-of-Words (BOW)

A bag-of-words representation is the one in which each SMS is represented by a vector of the size of the vocabulary |V|. The matrix composed of all these 5774 SMS which are input for our algorithms. It creates a unigram model of the text by keeping the number of occurrences of each word in a given document. Figure 2 shows the pseudo code of bag-of-words method.

2.2.2 TF-IDF (Term Frequency-Inverse Document Frequency)

If a word appears in several SMS, it is less representative of the SMS than a word that only appears in one SMS, TF-IDF reduces the number of occurrences of a word in the SMS by considering the number of all SMS in the dataset containing this word. Mathematical equations of TF * IDF are as follows [5]:

```
def  TF_IDF (bow_features):
  numrows = len(bow_features)
  numcols = len(bow_features [0])
  IDF= initialize the matrix with (numcls)
  TFIDF_Features= initialize the matrix with (numrows,numcols)
  N=Nbre of SMS in the dataset
  k=0
 for j in numcols:
    for i in numrows :
      if(bow_features [i][j]!=0):
        k+=1
    IDF[i][j]=log2(N divide k)
 for i in numrows :
    for j in numcols:
      if(bow_features [i][j]!=0):
        TF= bow_features [i][j] divide the sum of bow_features in a row i
        add the TF*IDF[i] to TFIDF_Features in (i,j)
      else:
        add 0 to TFIDF_Features in (i,j)
return TFIDF_Features
```

Fig. 3 TF-IDF pseudo code

$$\mathrm{TF}(i, j) = \frac{(\text{Term } i \text{ frequency in document } j)}{(\text{Total terms in document } j)} \tag{1}$$

$$\mathrm{DF}(i) = \log_2\left(\frac{\text{Total documents}}{\text{documents with term } i}\right) \tag{2}$$

Figure 3 shows the pseudo code of TF-IDF method.

2.3 Machine-Learning Classifiers

After extracting features, accuracy is being evaluated using four supervised machine-learning algorithms: Multilayer perceptron, support vector machines, random forest, and k-nearest neighbors. These algorithms are described as follows.

2.3.1 Multilayer Perceptron MLP

A multilayer perceptron is a deep learning technique, so it is a feed-forward network connecting at least three layers of nodes in an oriented graph, an input layer, a hidden layer, and an output layer. An MLP uses backpropagation as a supervised learning technique that must a set of training data with the relatively desired outputs to adjust the weights iteratively using the back propagated errors function of the errors returned. The input layer contains the inputs features of the SMS. The first hidden layer receives the weighted inputs from the input layer and sends data from

the previous layer to the next one. Finally, the output layer contains the classification result where the output '0' indicates non-spam and the output '1' indicates spam [7, 8].

The best value of parameters is {'Hidden_Layer_Size': 200, 'alpha': 0.001}.

Alpha is a parameter for regularization term.

The activation function for hidden layer is Relu, it returns

$$f(x) = \max(0, x) \tag{3}$$

The solver function by default is adaptive moment estimation (Adam) [8]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{4}$$

$$v_t = \beta_2 v_t - 1 + (1 - \beta_2) g_t^2 \tag{5}$$

where g_t denotes the gradient at time step t of objective function. m_t and v_t updates biased first moment estimate and second raw moment estimate of the gradients, respectively.

$\beta_1, \beta_2 \in [0, 1]$ presents exponential decay rates for the moment estimates.

Good default settings for the tested machine-learning problems are $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

So for Computing bias-corrected first moment estimate and second raw moment estimate:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1} \quad \text{and} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2} \tag{6}$$

To update parameters of function objective:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{V}_t} + \epsilon} \cdot \widehat{m}_t \tag{7}$$

2.3.2 Support Vector Machines SVM

Support vector machine is a supervised machine-learning approach in our work used for data classification. The objective of the support vector machine algorithm is to find the optimal hyperplane as the decision surface should maximize the distance between positive and negative data points. The radial basis function (RBF) kernel was preferred in this study due to its proven performance in text classification research before [8]. It is a reasonable first choice. This kernel nonlinearly maps samples into a higher dimensional space so it unlike the linear kernel case when the relation between

class labels and attributes is nonlinear. SVC classifier using an RBF kernel has two parameters [9, 10]: gamma and C.

The best value of parameters of RBF are {'C': 10, 'gamma': 0.1}

While C is a penalty parameter of the error term, it controls the trade off between classifying the training data correctly and maximizing the margin of the decision function.

Gamma is a parameter for non linear hyperplanes. The higher the gamma value it tries to exactly fit the training data point, the 'curve' of the decision boundary is high, which creates islands of decision-boundaries around data points.

The principal goal of SVM, is to produce a best model based on the training data, and can be able to predict a class label of the test data.

Radial Basis Function (RBF):

$$K\left(x_i, x_j\right) = \exp\left(-\gamma \|x_i - y_i\|^2\right), \quad \gamma \text{ is a Kernel parameter } \gamma > 0 \tag{8}$$

$$x_i, y_i \in R^m \quad i = 1, 2, \ldots, m$$

$||x_i - y_i||^2$ is the squared Euclidean distance between two feature vectors x_i, y_i.

2.3.3 Random Forest

Random forests is supervised learning algorithm, it is a combination of decision trees. Random forest builds several decision trees and merges them to get a more accurate and stable prediction, to classify a new object which is performed by each tree, the trees marks their votes for that class. The class having most number of votes decides the classification label 'Spam or Ham.'

The best value of random forest parameters: criterion = 'gini' [11], n_estimators = 31 n_estimators is the number of trees in the forest.

Criterion is the function to measure the quality of a split, this parameter is tree-specific.

$$I_G(f) = \left(1 - \sum_{i=1}^{m} f_i^2\right) \tag{9}$$

where m is the number of classes and f_i is the probability that a tuple in training dataset D belongs to class Ci.

2.3.4 K-Nearest Neighbors KNN

KNN is a simple supervised machine-learning algorithm, it used for classification and regression problems, it is a special type of algorithm that does not use a statistical model. It is 'non-parametric,' and it is based only on training data (feature vectors and labels). This type of algorithm is called memory-based algorithm. More specifically, the principle of this model consists to classify data point based on the labels of the k-nearest neighbor by the majority vote. The distance function for KNN is a Euclidean distance [8]:

$$\mathrm{d}(x, y) = \|x - y\| = \sqrt{(x - y) \cdot (x - y)} = \left(\sum_{i}^{m} (x_i - y_i)^2 \right)^{1/2} \tag{10}$$

$$x, y \in R^m \quad i = 1, 2, \ldots.., m$$

where $||x - y||$ is the Euclidean distance between two data vectors x, y.

In our work, the best value of K is 13.

2.3.5 Evaluation Metrics

To evaluate a spam detection system, we considered the standard metrics, true positive rate, true negative rate, false negative rate, false positive rate, precison, recall, F-measure, ROC curve, as described as follows (Fig. 4):

- True positive rate (TPR): The percentage which the machine-learning classifier predicts correctly the SMS as spam.
- True negative rate (TNR): The percentage which the machine-learning classifier predicts correctly the SMS as not spam.
- False negative rate (FNR): The percentage which the machine-learning classifier predicts the SMS spam as not spam.
- False positive rate (FPR): The percentage which the machine-learning classifier predicts the not-spam SMS as SMS spam.
- Precision: It means the correctness, the percentage of messages that the classification algorithm classified as spam and it were spam. It is given as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

Fig. 4 Confusion matrix

		SMS Spam	Not-Spam SMS
Predict	SMS as Spam	TP	FP
	SMS as not Spam	FN	TN

– Recall: It means the completeness, it calculates as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

– F-measure, it is the harmonic mean of precision and recall.

$$\text{F} - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall}).} \tag{13}$$

– Receiver operating characteristics (ROC) curve—is plotted between true positive rate and false positive rate for different threshold values.

3 Results and Discussions

In this paper, we have used python language, Google Colaboratory tensorflow-gpu as runtime environment.

Before evaluating our machine-learning algorithms, our dataset is a collection of SMS spam collected from mobile phone research. It is defined as follows.

The dataset of SMS spam has been created by Tiago A. Almeida and José María Gómez Hidalgo. It contains 5574 messages in English [2], extracted from free sources for research.

425 SMS from Grumbletext website, 3375 SMS from the NUS SMS Corpus (NSC), 450 SMS from Caroline Tag's PhD Thesis and 1324 SMS from the SMS Spam Corpus v.0.1 Big. It contains a collection of 747 SMS spam and 4827 legetimate messages (ham). The dataset is a large text file contains one message per line. Each line is composed by two columns: column 1 contains the label (ham or spam) and column 2 contains the raw text.

After pre-processing and extracting the features, our aim is to search the best classifier for detecting the SMS spam. To choose the best parameters for each classifier, the GridsearchCV() function from scikit-learn library is implemented. We have used cross validation of tenfold in which 75% of data is used for training and 25% of data is used for testing the different model of machine-learning algorithms.

Figure 5 shows the accuracy scores using the BOW and the TF-IDF methods.

Figure 5 shows that MLP and SVM classifiers have the higher accuracy with the bag-of-words method comparing to random forest and KNN.

Figures 6 and 7 shows the receiver operating characteristics (ROC) curve for our proposed algorithms.

AUC—ROC curve is the area under the curve.

MLP and SVM have the same value of AUC-ROC in each figure. The higher values of AUC-ROC are, respectively, 94% in Fig. 6 and 93% in Fig. 7. These model classifiers are almost perfectly capable to distinguish between SMS spam and legitimate SMS with value of 94% and 93% (Table 1).

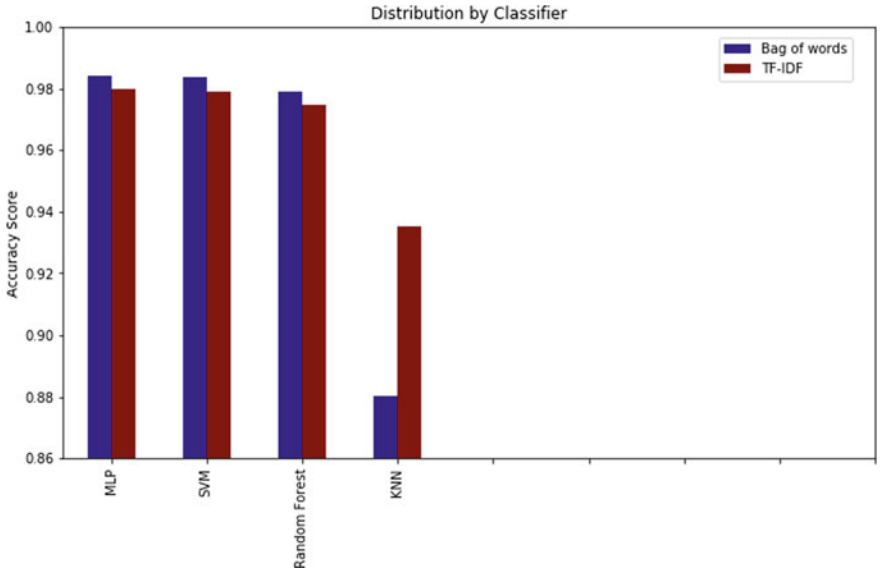

Fig. 5 Machine-learning classifiers performance

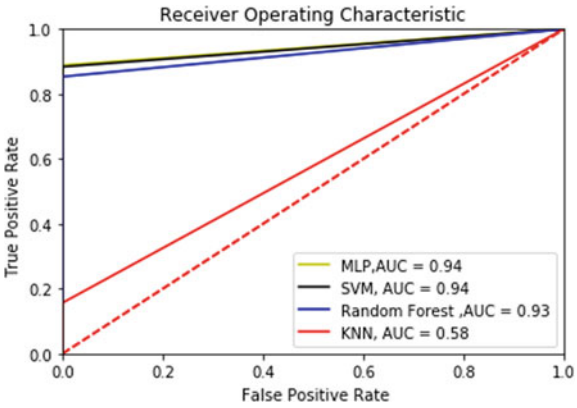

Fig. 6 ROC curve of ML. Classifiers using bow model

For bag-of-words method, KNN classifier shows the least precision rate 96.48%, the least recall rate 15.65%, and the least f-measure 56.06%, also a lower accuracy with value 88.01%. On the other hand, the MLP shows the best classification results with a higher accuracy of 98.42%, recall rate of 88.88%, precision rate of 99.37%, and f-measure of 94.11%.

For TF-IDF method, KNN algorithm improves their results of accuracy value 93.53% instead of 88.01%, recall rate of 56.21%, precision rate of 98.26% and f-measure of 71.51%; however, it shows usually the less results. While the MLP has

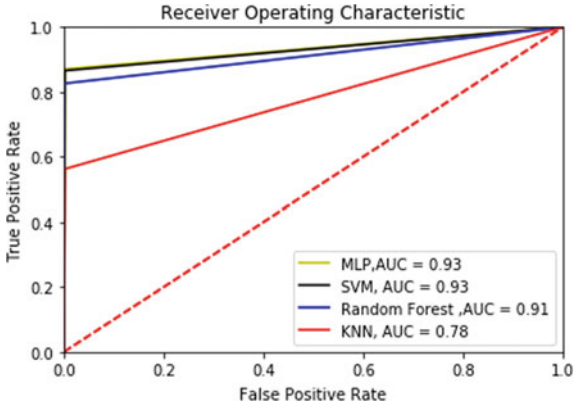

Fig. 7 ROC curve of ML. Classifiers using TF-IDF model

Table 1 Performance metrics of classifiers using bow and TF-IDF models

	Accuracy		Recall		Precision		F-measure	
	Bow (%)	TF-IDF (%)	Bow (%)	TF-IDF (%)	Bow (%)	TF-IDF (%)	Bow (%)	TF-IDF (%)
MLP	98.42	97.98	88.88	87.06	99.37	98.87	94.11	92.59
SVM	98.34	97.91	88.38	86.56	99.1	98.86	93.83	92.3
R. forest	97.91	97.48	85.35	82.58	99.31	98.7	90.09	90.46
KNN	88.01	93.53	15.65	56.21	96.48	98.26	56.06	71.51

the best accuracy of 97.98%, often the best recall, precision, and f-measure of values, respectively, 87.06%, 98.87% and 92.59%.

The MLP has the best results using the two methods of extracting features comparing to the other classifiers, but using the bag-of-words method, the MLP achieves the best accuracy (98.42% instead of 97.98%), recall (88.88% instead of 87.06%), precision (99.37% instead of 98.87%), and the best f-measure (94.11% instead of 92.59%).

Taking into consideration the results obtained in our work and comparing it with the paper [4], listed in introduction, MLP has produce better precison using the bag-of-words method on the same dataset. It shows that MLP can be useful for SMS spam detection and similar classification tasks.

4 Conclusion

Various algorithms of machine-learning MLP, SVM, random forest, KNN, was implemented to provide the best classifier using the bag-of-word and the TF-IDF methods, applied to the Tiago's SMS spam dataset. The results show that the multilayer perceptron achieves the best results comparing to the other classifiers with accuracy of 98.42%.

In our future work we will try to use other machine-learning classifiers, in order to give a best accuracy and a best value of AUC-ROC.

References

1. Gupta, M., Bakliwal, A., Agarwal, S., Mehndiratta, P.: A comparative study of spam SMS detection using machine learning classifiers. In: 2018 Eleventh International Conference on Contemporary Computing (IC3), pp. 287–293
2. SMS Spam Dataset 'Collection V.1'. Available online at http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/
3. Spam SMS Dataset 2011–12. Available online on request at http://precog.iiitd.edu.in/requester.php?dataset=smsspam
4. Choudhary, N., Jain, A.K.: Towards filtering of SMS spam messages using machine learning based technique. In: Advanced Informatics for Computing Research: First International Conference, ICAICR 2017, Jalandhar, pp. 18–30, 17–18 Mar 2017 (Revised selected papers)
5. Uysal, A.K. Gunal, S. Ergin, S. Gunal, E.S.: The impact of feature extraction and selection on SMS spam filtering. 2013 Elektronika Ir Elektrotechnika, 67–72
6. Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: Convolutional neural network based SMS spam detection. In: 26th Telecommunications Forum TELFOR 2018, pp. 807–810
7. Goh, K.L., Lim, K.H., Singh, A.K.: Multilayer perceptrons neural network based Weh spam detection application. In:2013 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Beijing, China, pp. 636–640
8. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: ICLR 2015 Conference, pp 34–48
9. Kim, J., Kim, B., Savarese, S.: Comparing image classification methods: K-nearest-neighbor and support-vector-machines. In: American-Math'12/CEA'12 Proceedings of the 6th WSEAS International Conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics, pp. 133–138
10. Hsu, C.-W., et al.: A practical guide to support vector classification. Available at https://www.csie.ntu.edu.tw/~cjlin (2016)
11. Sedhai, S., Sun, A.: Semi-supervised spam detection in twitter stream. IEEE Trans. Comput. Soc. Syst., 169–175 (2018)