

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325270587>


Ham and Spam E–Mails Classification Using Machine Learning Techniques

Article in Journal of Applied Security Research · May 2018
DOI: 10.1080/19361610.2018.1463136

CITATIONS
20

READS
5,325

3 authors:



Mahmoud Bassiouni

Egyptian E-Learning University

9 PUBLICATIONS 76 CITATIONS

SEE PROFILE




Mayar Aly Shafaey

National Egyptian E-Learning University

10 PUBLICATIONS 46 CITATIONS

SEE PROFILE




El-Sayed A. El-Dahshan

Ain Shams University


72 PUBLICATIONS 1,463 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Signal Processing [View project](#)



Machine Learning Approaches for biometrics and diagnosis [View project](#)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325270587>

Ham and Spam E-Mails Classification Using Machine Learning Techniques

Article in Journal of Applied Security Research · May 2018
DOI: 10.1080/19361610.2018.1463136

CITATIONS
0

READS
42

3 authors:



Mahmoud Bassiouni
Egyptian E-Learning University
7 PUBLICATIONS 6 CITATIONS

SEE PROFILE



Mayar Aly Shafaey
Egyptian E-Learning University
4 PUBLICATIONS 0 CITATIONS

SEE PROFILE



El-Sayed Ahmed El-Dahshan
Ain Shams University
50 PUBLICATIONS 452 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Signal Processing [View project](#)



PCG signal processing [View project](#)



Ham and Spam E-Mails Classification Using Machine Learning Techniques

M. Bassiouni, M. Ali & E. A. El-Dahshan

To cite this article: M. Bassiouni, M. Ali & E. A. El-Dahshan (2018) Ham and Spam E-Mails Classification Using Machine Learning Techniques, Journal of Applied Security Research, 13:3, 315-331, DOI: [10.1080/19361610.2018.1463136](https://doi.org/10.1080/19361610.2018.1463136)

To link to this article: <https://doi.org/10.1080/19361610.2018.1463136>



Published online: 21 May 2018.



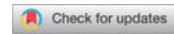
Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)



Ham and Spam E-Mails Classification Using Machine Learning Techniques

M. Bassiouni^a, M. Ali^b, and E. A. El-Dahshan^{a,b}

^aEgyptian E-Learning University (EELU), Eldoki, El-Geiza, Egypt; ^bFaculty of Science, Department of physics, Ain Shams University, Abbassia, Cairo, Egypt

ABSTRACT

Spam e-mail has become a very serious problem. Sending inappropriate messages to a large number of recipients indiscriminately has resulted in anger by users but large profits for spammers. This article looks at classifying spam e-mails from inboxes. Ten alternative classifiers are applied on one benchmark dataset to evaluate which classifier gives better result. A 10-fold cross validation is used to provide the accuracy. Results of the classification algorithms are compared with the spambase UCI dataset. The experimental results approve that the spam mails can be classified correctly, with accuracy reaching up to 95.45% for the Random Forest technique, compared to other classifiers used.

KEYWORDS

Spam; e-mail classification; machine learning algorithms; feature selection; spambase

Introduction

Nowadays, most people have access to the Internet, and they cannot survive without Smartphone and computers. They not only use the Internet for fun and entertainment, but they also use it for business, stock marketing, searching, sending e-mails, and so on. Hence, the usage of the Internet is growing rapidly (Christina, Karpagavalli, & Suganya, 2010). One of the threats for such technology is a spam. Spam has a lot of definitions, as it is considered one of the complex problems in e-mail services. Spam is a junk mail/message, or an unsolicited mail/message. Spam e-mails are also those unwanted, unsolicited e-mails that are not intended for a specific receiver. It is basically an online communication sent to the user without permission. It takes on various forms like adult content, selling products or services, job offers, and so forth (Zhang, Zhu, & Yao, 2004). The spam has increased tremendously in the last few years. The good, perfect, and official mails are known as *ham*. It is also defined as an e-mail that is generally desired.

Today, more than 85% of mails or messages received by users are spam (Manisha & Jain, 2015). It costs the sender very little time to send, but most of the costs are paid by the recipient or the service providers rather than by the sender. The cost of

CONTACT M. Bassiouni ✉ mbassiouni@eelu.edu.eg 📍 Egyptian E-Learning University (EELU), 33 El-messah Street, Eldoki, El-Geiza 11261, Egypt.

Color versions of one or more of the figures in this article can be found online at www.tandfonline.com/wasr.

© 2018 Taylor & Francis Group, LLC

spam can also be measured in lost human time, lost server time and loss of valuable mail/messages (Alsmadi & Alhami, 2015). The sent mail reserves a quota in the server, and the receiver may have a limited space, causing the server to reject another ham mail because it is out of space. Moreover, the reader may lose a lot of time in reading unuseful messages. They can also drain resources; such as bandwidth, storage capacity, and productivity loss, and interfere with the expedient delivery of valid mails (Yu & Xu, 2008).

Therefore, the problem that is addressed in this article is how to develop an automated method for classifying the spam mails out of inbox ones (ham). Furthermore, this article addresses how to reach a high performance with low computation for real-time process. The Machine Learning field has a robust, ready-made and alternative way for solving this type of the problem (Awad & Elseuofi, 2011). The article is organized as follows: The second section gives a survey of known approaches used for classification process, using different machine learning techniques. The third section represents the proposed method in detail, and the fourth section shows the experimental results with scientific comments. Finally, some conclusions are presented in the fifth section for highlighting the important results.

Related work

The effective publications for spam e-mails classification were investigated in a wide variety of articles. In Table 1, a survey is given of the techniques which were applied for e-mails classification and filtration. Sharaff, Nagwani, and Dhadse (2016) used a machine learning algorithm to classify ham and spam. They applied four experiments for classification, using WEKA software (Hall et al., 2009). The dataset used for testing the four classifiers was based on Enron. Those classifiers are (Iterative Dichotomiser) ID3, Decision Tree (J48), Simple Cart and Active directory Tree (AD tree). The accuracies of each classifier were (92.7%) for J48, (89.1%) for ID3, (90.9%) for AD Tree and (92.6%) for simple cart. (Scholar, 2010) worked on different classifiers to determine whether the mail is ham or spam. The data used for experimenting the classifiers was separated into two parts; one part was used for training and the other for testing. The testing data was done using 10-fold cross validation method. Three classifiers were applied from WEKA software: The J48, Multilayer perceptron (MLP) and simple logistic. Their accuracies were 93%, 92%, and 92% respectively. Other three classifiers were used from rapid miner tool: The Naïve Bayes (NB), MLP, and linear discernment analysis (LDA). Their accuracies were 90%, 93%, and 92% respectively. It is observed that MLP were top performers in all cases, and thus can be deemed consistent. Youn and Mcleod (2007) identified a system for classifying the e-mail data. The system was based on four classifiers, different database size, and different parameters. The maximum number of e-mails used in training was 4,500 different e-mails; into 38.1% of spam and 61.9% of ham. Those four classifiers were based on neural network (NN), Support vector Machine (SVM), NB, and J48. The accuracies were based on the number of e-mails and number of features in each e-mail. The data size of e-mails ranged from 1,000

Table 1. A survey of the machine learning techniques for spam mails classification.

Author(s)	Methodologies			Results
	Application	Dataset	Machine learning algorithms	
Sharaff et al., 2016 Scholar, 2010 Youn & Mcleod, 2007	Classification algorithms for spam e-mail detection Spam e-mail classification Spam e-mails classification	Enron — Size of 1000, size of 5000	Probabilistic, Decision Tree, and Vector Machines Multilayered Perceptron (MLP), and J48 SVM, NB, and J48.	J48 and Bayes Net algorithms perform better than Support vector machines (SVM) 93% for J48, slightly exceeding MLP's 92%, In case of 1000: SVM, NB, and J48 had 92.7%, 97.2%, and 95.8% respectively. In case of 5000: SVM accuracy dropped by 1.8% and NB by 0.7%, while J48 increased by 1.8%. Accuracy = 99.5%
Chen et al., 2009	Spam e-mail filtration	400 mails for Test 200 for Train	Support Vector Machines	In case of 400: 90%.
Provost, 1999	Identifying spam e-mails	400 mails and 50 mails	Repeated Incremental Pruning to Produce Error Reduction (RIPPER)	In case of 50: 95%. Accuracy = 96.4% using EDT on Spambase
Kiran & Atmosukarto, 2009	Classification for spam e-mails	Spambase	Five experiments	Accuracy = 96.23% using NB on SpamAssassin Random Committee achieved the best result with 94.28% accuracy
Sharma & Arora, 2013	Spam e-mail identification	SpamAssassin Spambase UCI	24 classifiers from the WEKA toolset	Two-stage smoothing version was the highest accuracy Accuracy = 93.1%
Kaur & Bogiri, 2014 Awad & Foqoha, 2016	E-mail spam filtration system Spam e-mail classification	Enron and fresh spam messages Spambase UCI	Five versions of NB Radial Basis Function Particle Swarm Optimization (RBFPSO)	Bagging of trees gave best result whereas SVM gave worst results. PCDAR proved to be better than SVM
Silva et al., 2012	Spam e-mail classification	WEB SPAM UK 2006	Link based and transformed link based for features and bagging of tree and SVM for classification	
Gomez & Moens, 2012	Spam e-mail classification	PU1, Ling Spam, Spam Assassin, Phishing and TREC7 spam corpus.	Principal Component Analysis Document Reconstruction (PCADR) and SVM for classification	
Kumar et al., 2012	Spam e-mail filtering	Spambase UCI	Fisher filtering, Relief, Runs Filtering and Step disc for feature selection and Tree for classification	Accuracy = 99%
Bhat et al., 2014 Trivedi & Dey, 2013	Spam e-mail classification Spam e-mail classification	Data Set is taken from Facebook Enron Dataset	Base Classifiers (J48, IBK, Naïve Bayes) Genetic Search algorithm to select most important features 134 out of 1359 Bayesian and Naïve Bayes for classification	J48 has performed better than others Bayesian Classifier has given best accuracy = 92.9%

Note. NB = The Naïve Bayes; DT = decision tree; VM = vector machines; BN = Bayes Net; MLP = multi-layered perceptron; SVM = support vector machine; RIPPER = repeated incremental pruning to produce error reduction; EDT = ensemble decision table; PCADR = principal component analysis document reconstruction; IBK = instance based k -nearest neighbor.

to 4,000, and the features ranged from 10 to 55. Chen, Liu, Zhu, and Qiu (2009) discussed the problem of spam filtering, based on weighted SVM. The dataset was divided into three groups, and the highest accuracy was from the second group with 99.44%. The experimental result shows that the weighted SVM reduces the degree of misclassification of legitimate e-mails effectively, while the classification accuracy is reduced a little. Provost (1999) proposed an algorithm based on NB with bag-valued features and the repeated incremental pruning to produce error reduction (RIPPER) rule-learning algorithm. The RIPPER showed a promising performance in text categorization task. NB is recognizing spam with 90% accuracy after training on 25 examples and has reached 95% accuracy after 50 examples, while RIPPER reached 90% accuracy after 400 training samples. Kiran & Atmosukarto (2009) used eight classifiers in order to identify whether the e-mail is ham or spam. He applied five experiments in order to test the performance using ensemble decision tree (EDT), NB, and complement Naïve Bayes algorithms. They reached about 96.4%, 96.23%, and 81.2%, *respectively*. It is concluded from those various experiments that the attributes that are statistically meaningful have a dense presence over the entire data; as they tend to be data-independent and can provide better separability over many kinds of spam and ham. Sharma and Arora (2013) presented 24 classifiers to classify the ham and spam from the WEKA software. They worked on the spambase UCI dataset, and the highest accuracy reached by using 10-fold cross validation was random committee with 94.37%. Kaur and Bogiri (2014) proposed different versions of NB and compared them on six new, nonencoded datasets. Those datasets contain ham messages of particular Enron users and fresh spam messages. Those versions were based on (JM) smoothing, Dirichlet smoothing, absolute discounting, and two-stage smoothing. They conclude that the two-stage smoothing performs well with NB and that it was the highest performance of them all on the six datasets. It is concluded that NB performs much better for different data collections than other classifiers. NB also suffers from some issues; like unseen words. So, different smoothing techniques were used, and two-stage smoothing performs well with NB. Awad and Foqoha (2016) proposed a combination of Radial bias function neural network (RBFNN) and Particle swarm optimization (PSO) to be known as (HC-RBFPSO). They worked on spambase dataset with 70% for training and 30% for testing. The performance was measured using different values of neurons in the hidden layer ranging from 10 to 50. The maximum accuracy achieved was 93.1%. It is concluded that the hybrid approach between RBF that is characterized by better approximation and PSO to find optimal centers of hidden neuron show a better performance in terms of accuracy. Silva, Yamakami, and Almeida (2012) worked on classifying the ham and spam based on different feature sets. Those feature sets are based on content, linked and transformed-linked based combination of content and link-based. The dataset used was webspam UK for classification. Eight classifiers were used to test the performance of those features. Content-based features with bagging classifier gave the highest accuracy. It is concluded from the experiments that since the data is unbalanced, evaluated techniques are made superior when trained with the same number of samples of each class. This is because the

models are biased to the benefit of the class with the largest number of samples. Gomez and Moens (2012) introduced two classifiers to verify ham and spam e-mail. Those classifiers are based on principle component analysis document reconstruction (PCADR), based on power factorization method and SVM. The public e-mail corpora were used for performing the test with 1,250 ham and 1,250 spam. The results showed that PCADR was better than SVM with 93.67% accuracy, using 10-fold cross validation, and much faster than SVM. It is concluded that PCADR is competitive in training time, in comparison to SVM. Furthermore, it is considered well suited for classification when training with a labeled dataset collected by using a given setup, and when testing with a dataset collected with another setup. Kumar, Poonkuzhali, and Sudhakar (2012) worked on different classifiers to classify ham and spam data. They worked on the spambase UCI dataset and used different feature selection filtering techniques. The best results achieved were by using random tree with ffisher filtering and run time filtering. The accuracy reached was more than 99%. Bhat, Abulaish, and Mirza (2014) proposed ensemble learning approaches based on bagging, boosting, and stacking. The data was collected from the facebook for classification. Experimental results reveal that the bagging ensemble learning approach, using J48 (decision tree) base classifier, performs better than its individual model, and is better than some other ensemble learning approaches for spammer detection, using structural social network features. Trivedi and Dey (2013) consider two probabilistic algorithms based on Bayesian and NB, and three boosting algorithms; such as Bagging, Boosting with Resampling, and AdaBoost. Initially, the Probabilistic classifiers were tested on the Enron Dataset without Boosting and, thereafter, with the help of Boosting algorithms. The Genetic Search Method was used for selecting the most informative 375 features out of 1,359 features created at the outset. The results showed that in identifying complex Spam messages, Bayesian classifier performs better than NB with or without boosting. It is conducted that the boosting algorithms play an important role in the performance of classifier.

Based on the previous survey, we conclude some important issues. First, it presented wide and effective techniques for ham and spam classification. Second, different datasets were introduced for that purpose and carried out on different number of instances. Third, different classifiers and preprocessing, as well as feature selection techniques were used for ham and spam classification. Definitely, the proposed work will follow the aforementioned approaches, taking into consideration the different features of our mails. Finally, ham and spam classification is considered to be an open-research question.

Proposed methodology

Our proposed methodology consists of four main stages. The first stage is the data acquisition, in which we collect data from a database called spambase UCI. In the second stage, we apply preprocessing step on the data using normalization. In the third stage, we obtain the features using Infinite latent feature selection. Finally, 10

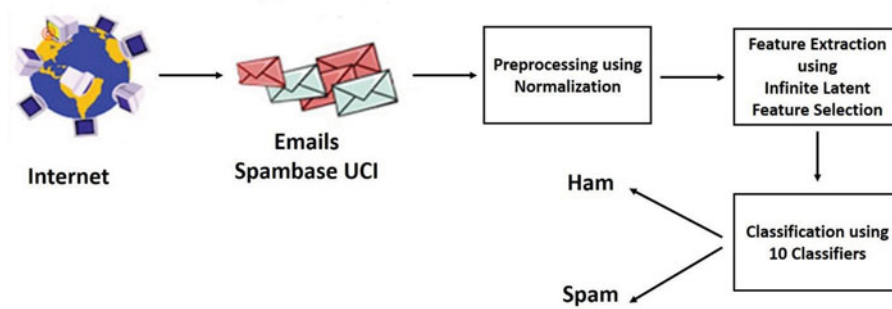


Figure 1. Our proposed methodology for Spambase UCI dataset classification.

classifiers are used to identify whether the features are of a ham or spam as shown in Figure 1.

Dataset

There is a dataset called *spambase* UCI which is used to evaluate the proposed method presented here. It contains a set of spam and nonspam classes. This dataset consists of 4,601 number of instances and 57 number of attributes. The last column of this database denotes whether the e-mail is considered spam (1) or not (0). Most of the attributes indicate whether a particular word or character is frequently occurring in the e-mail or not. The runlength attributes (55–57) measure the length of sequences of consecutive capital letters. Here are the definitions of the attributes, see Table 2:

Preprocessing

Most of the data in the real word are inadequate, as they contain a lot of noise and missing values. Any efficient decision depends on an efficient mining on a

Table 2. The definition of the spam mails attributes.

# Attributes	Data type	Interval	Description
48	Continuous real	[0,100]	= percentage of words in the e-mail that match WORD, i.e., $100 * (\text{number of times the WORD appears in the e-mail}) / \text{total number of words in e-mail}$. A “word” in this case is any string of alphanumeric characters bounded by nonalphanumeric characters or end-of-string.
6	Continuous real	[0,100]	= percentage of characters in the e-mail that match CHAR, i.e., $100 * (\text{number of CHAR occurrences}) / \text{total characters in e-mail}$.
1	Continuous real	—	= average length of uninterrupted sequences of capital letters.
1	Continuous integer	—	= length of longest uninterrupted sequence of capital letters.
1	Continuous integer	—	= sum of length of uninterrupted sequences of capital letters = total number of capital letters in the e-mail.
1	Nominal	{0,1}	= denotes whether the e-mail was considered spam (1) or not (0), i.e., unsolicited commercial e-mail.

good quality of data. So, preprocessing is one of the major tasks for data. Data cleaning, integration, transformation and reduction are from those major tasks in preprocessing. In this dataset, data normalization is done before performing any further processing.

Feature selection

Our feature selection is based on an Infinite latent feature selection (ILFS) Roffo, Melzi, Castellani, and Vinciarelli (2017). A training set X represented as a set of feature distributions $X = \{X_1, \dots, X_n\}$ is given, where each $m \times 1$ vector X_i is considered the distribution of the values assumed by i th feature with regard to the m samples. We build an undirected graph G , where nodes a_{ij} correspond to features and edges model relationship between any pairs of nodes. Let an adjacency matrix A associated to G define the nature the weighted edges. The weighted edges model the relationship between features. Each weight represents the likelihood that features X_i and X_j are good candidates. Weights can be associated to a binary function of the grade nodes:

$$a_{ij} = \varphi (X_i, X_j) \quad (1)$$

Where $\varphi (\cdot, \cdot)$ is considered to be a real-valued function learned by the probability of each co-occurrence in X_i, X_j , as a mixture of an independent multinomial distributions. Considering a weighted graph G , ILFS represents the subsets of features as paths joining them. The weight of each path that represents the features is known by the joint probability of all nodes fitting to it. ILFS feats the convergence property of the power series of matrices, and evaluates it in an elegant form with weight of each feature with respect to all the others joined together. In the end, a set of ranking scores are obtained based on these weights. Those rankings show the most discernment features in the 57 features and order them for classification.

Classification

Based on previous literature, we realize that the machine learning techniques play a major and important role in the data classification applications.

Data classification is defined as a process in which individual items are grouped based on the similarity between the data and the description of the group (Mohd, Yuk, Wei-Chang, Noorhaniza, & Ahmad, 2011).

The following subsections have a detailed description for each technique (classifier) we used. In this method presented here, the classification process is based on 10 classifiers listed as follows: Random forest (RF), Artificial Neural network (ANN), Logistic, SVM, Random Tree, K-nearest neighbor (KNN), Decision Table, Bayes Net, NB, and Radial Basis Function (RBF).

Random forest (RF)

Random Forest is considered one of the most recent classifiers. It grows many classification trees for verification or identification. In order to classify a new object from an input vector, this input is put down each of the trees in the forest. Each tree gives a classification, and the tree votes for that class. The forest takes the classification having the most votes (Rodriguez-Galiano, Ghimire, Rogan, Chica-Olmo, & Rigol-Sanchez, 2012).

Many learning algorithms (e.g. RF, bagging, and boosting) have received a great interest. They are more accurate and robust to noise than other single classifiers. Breiman suggested a new classifier 2001 called Random forest, which presents a lot of advantages especially in classification:

- It can handle thousands of input features without any deletion of features.
- It shows an approximation of which features are important in the process of classification
- It can produce an internal unbiased approximation of the generalization error.
- It computes contiguities between pairs of cases that are used in positioning outliers.
- RF are very robust to outliers and noise
- The complexity of RF is lighter than any other classification methods, and it is computationally lighter than other tree collaborative methods.
- A RF consists of a combination of classifiers, where each classifier contributes with a single vote to the assignation of the most frequent class to the input vector (x),

$$C_{rf}^B = \text{majority vote } \{C_b(x)\}_1^B \quad (2)$$

where C_{rf}^B is the class prediction of the bth random forest tree. RF is considered to be a grouping of many classifiers, and it consists of some characteristics that make it more efficient than other traditional classification trees. RF increases the diversity of the trees by letting them grow, using different training data subsets that employ the concept of bagging or bootstrapping.

Artificial neural networks (ANN)

ANN is considered one of the important classifiers in machine learning. Its classification operation begins or starts with the sum of multiplication of weights and inputs adding the bias at the neuron. If this summation is positive then only output elements fire. Otherwise it doesn't fire. ANN is a machine learning adaptive system, in which the system adapts itself and changes its weights during each iteration (Haykin, 2008).

The input vectors are applied to the two layers feed forward neural network and are trained with back propagation algorithm. Training means adapting the weights of the network and the weights are updated until the operation gives the desired output. The equation that represents the interval activity of the neuron is given in

the following formulas (2) and (3):

$$Z_j = f \left(\sum_{i=1}^d w_{ji} x_i + w_{j0} \right) \quad (3)$$

$$Y_k = f \left(\sum_{j=1}^{n_k} w_{kj} Z_j \right) + w_{k0} \quad (4)$$

Where x_i the input vector, w_{ji} is the weight between the input and the hidden layer, w_{j0} is the bias of the hidden layer neurons, Z_j is considered to the output of the hidden layers. Y_k is the output of the network, w_{kj} is the weight between the output and the hidden layer, w_{k0} is the bias of the neurons of the output layer.

Logistic regression (LR)

LR is a linear classifier and has an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits:

$$\frac{1}{(1 + e^{-value})} \quad (5)$$

Input features (x) are combined linearly using weights or coefficient values to predict an output value (y) (Yan & Lee, 2005):

$$Y = \frac{e^{(b_0 + b_1 * x)}}{(1 + e^{(b_0 + b_1 * x)})} \quad (6)$$

Where y is the predicted output, b_0 is the bias or intercept term and b_1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient that must be learned from the training data. The main aim of the logistic is to determine the best fitting model and to describe the relationship between the categorical representatives of dependent variable.

Support vector machine (SVM)

SVM is an approach that is used for pattern classification; it is based on a statistical theory proposed by Vapnik (Vapnik, 2013). SVM works well for both linearly separable features and nonlinearly separable features. The overall problem comes down to finding a function that reduces the error and successfully classifies the input features.

The advantage of SVM is the ability to classify tested input features with high accuracy, as it works on the concept of maximum margin hyperplane. The classifier performances have been improved for small sample learning problems by applying Sequential Minimal Optimization (SMO) and polynomial kernel function.

Our Experimental results indicate that SVMs can accomplish a performance that is greater than or equal to other classifiers, while requiring significantly less training data to achieve such an outcome.

The SVM decision function is defined as follows:

$$F(y) = \sum_{i=1}^N \alpha_i K(x_i, y) + b \quad (7)$$

Where y the unclassified tested feature, x_i is are the support vectors and α_i their weights, and b is a constant bias. $K(x_i, y)$ is the kernel function which performs implicit mapping into a high-dimensional feature space. The support vectors are produced from the training samples through an optimization process. To train SVM, the solution of a large quadratic programming (QP) optimization problem is required. SMO breaks this QP problem into a series of small possible QP problems. Those problems are solved analytically, thus avoiding time-consumption. The memory required only for SMO is linear in the training size, which allows SMO to handle very large training. SMO balances somewhere between linear and quadratic in the training set size for various test cases (Platt, 1999).

Random tree

Leo Breiman and Adele Cutler introduced Random trees (Prasad, Iverson, & Liaw, 2006). It usually refers to built random trees. Those trees are trained with same parameters, but on different training sets. These sets are generated from the original training dataset, using bootstrap. For each training set, same number of vectors is randomly selected as in the original set. Those vectors are chosen with replacement, some vectors will occur more than once and some will be absent.

Not all variables are used to find the best split at each node of each trained tree, but a random subset of them. A new subset is generated with each node. Its size is fixed for all nodes and all trees. WEKA uses the term “random trees” to refer to a decision tree built on the random columns.

K-Nearest neighbor (KNN)

KNN is a classification method based on the closest training samples. It compares new tested features stored in memory with instances seen in training, instead of performing explicit generalization. KNN is an instance-based training algorithm because it constructs hypothesis directly from the training instances (Araújo, Nunes, Gamboa, & Fred, 2015).

All the samples found in the training set have their own value. We compare the test samples with all other samples already present in the training set. There are different methods for comparing these values; like Hamming distance, Euclidean distance, . . . and so forth. We choose the Euclidean distance for comparing between the test samples and the training samples sets. By comparing all the distances, we find the nearest neighbors that have the minimum Euclidean distance.

Let C be a test set described with parameters as $[C_{1,1}, C_{1,2}, C_{1,3} \dots, C_{1,k}]$ and T be a training set described as $[T_{1,1}, T_{1,2}, T_{1,3} \dots, T_{1,k}]$. The Euclidean distance between these two samples N and M is defined in the following equation:

$$D(C, T) = |C - T| \quad (8)$$

Decision table

Let it be supposed that we have training samples containing instances with its classes labeled. A hypothesis in some representation is built using an induction algorithm. The representation we investigate is a decision table with a default rule mapping to the majority class. It is sometimes called decision table majority DTM. A DTM has two components:

- Set of features (Schema)
- Multiset of labeled instances (Body)

Each instance contains a value for each of the features and a value for the label. Suppose an unlabeled instance I ; the label assigned to this instance by DTM classifier, is produced as follows. Let L be the set of labeled instances in the DTM, exactly matching the given instance I , where the features in the schema that are required to match are kept and the other features are ignored. If $L = \phi$, return the majority class in the DTM; otherwise return the majority class in L . The unknown values are preserved as different values in the matching process (Kohavi, 1995).

Bayes classifier

Classes are assigned in Bayesian classifiers (Rish, 2001) to a given example described by its feature vector. Those classifiers are learning with the assumption that features are an independent class; that is,

$$P(X|C) = \prod_{i=1}^n P(X_i|C) \quad (9)$$

Where $X = (X_1, \dots, X_n)$ is a feature vector and C is a class. Despite this unrealistic assumption, the resulting classifier is known as Bayes Classifier.

Radial basis function (RBF)

This architecture has a great advantage of computing speed compared to multiple hidden layer nets. Each hidden node in the RBF net defines one of the kernel functions. The output node is computed using the weighted sum of the hidden node outputs. The kernel function is a local function and the range of its effect is determined by the center and width. Its output is high when the input features are close to the center, and it is reduced to zero as the inputs distance from the center starts to increase. A popular kernel function is the Gaussian function, and it will be used in the algorithm of RBF (Roy, Govil, & Miranda, 1995).

The design and training of RBF net consists of:

- Knowing how many kernel functions can be used,
- finding the approximate centers and width, and
- finding the optimal weights that connect them to the output node.

In this method, a specific subset of the hidden nodes, related to class k , is linked to the k^{th} output node. The class k hidden nodes are not linked to the other output

nodes. Therefore, mathematically, the input $F_k(x)$ to the k^{th} output node is given by:

$$F_k(x) = \sum_{q=1}^{Q^k} h_q^k G_q^k(x) \quad (10)$$

$$G_q^k(x) = R(||x - c_q^k||) w_q^k \quad (11)$$

Here, Q^k is the number of hidden nodes associated with class k , q refers to the q^{th} class k hidden node, $G_q^k(x)$ is the response function of the q^{th} hidden node for class k , R is a radially symmetric kernel function, $c_q^k = (c_{q1}^k \dots c_{qn}^k)$ and w_q^k are the center and width of the q^{th} kernel function for class k , and h_q^k is the weight connecting the q^{th} hidden node for class k to the k^{th} output node.

Naïve bayas

It is a simple probabilistic classifier which operates based on Bayes theorem with powerful “naïve” independence resolutions, as in following equation:

$$P(c|x) = P(x|c) P(c) / P(x) \quad (11)$$

Where $P(c|x)$ is the posterior probability of class (c , target) given predictor (x , attributes), $P(c)$ is the prior probability of class, $P(x|c)$ is the likelihood which is the probability of predictor given class, and $P(x)$ is the prior probability of predictor.

Naïve Bayes classifier was proposed for phishing e-mail filtering in Microsoft (Kaur & Oberai, 2014) in 2006. Naïve Bayes are considered the individual attributes that are conditionally independent of each other in classification.

Results and discussion

To investigate the performance on the selected classification methods, namely; Random Forest, ANN, Logistic, SVM, Random Tree, KNN, Decision Table, Bayes Net, Naives Bayes, and RBF, we use the same experiment procedure as suggested by WEKA classification.

In WEKA, all data is considered as instances and features in the data, known as attributes. For experimentation, the UCI *spambase* dataset is used with a total of 4,601 data instances. Results of the simulation with True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), adding TP rate and FP rate, are shown in Table 3.

TP: Total number of spam e-mail correctly classified as ham.

TN: Total number of ham e-mails correctly classified as spam.

FP: Total number of ham e-mails misclassified as ham.

FN: Total number of spam e-mails misclassified as spam.

$$TP\ rate = \frac{TP}{TP + FN} \quad (12)$$

Table 3. Parameters of each of the classifiers used in identification of Spambase UCI dataset.

Classifier	Parameters	TP	FP	FN	TN	TP_rate	FP_rate
Random Forest	Trees = 100, Seed = 1	2713	75	134	1679	0.955	0.055
ANN	Iterations = 2000, lr = 0.3, mc = 0.2	2618	170	179	1634	0.924	0.084
Logistic	Max iterations = -1, ridge = 1*10 ⁻⁸	2645	143	206	1607	0.924	0.089
SVM	Kernel function = Polynomial	2651	137	236	1577	0.919	0.098
Random Tree	Min weight of instances in a leaf = 2	2592	196	191	1622	0.916	0.092
KNN	Linear Search with neighbors = 1	2585	203	221	1592	0.908	0.103
Decision Table	Based on Best First search	2663	125	321	1492	0.903	0.125
Bayes Net	Bayes search + Simple estimator	2620	168	301	1512	0.898	0.124
Naïve Bayes	Uses unsupervised Discretization	2621	167	300	1513	0.899	0.124
RBF	Cluster number = 2	2169	619	181	1632	0.826	0.148

Note. lr = Learning rate; mc : momentum constant.

Table 4. Seven measurements for spambase UCI classification.

Classifier	Precision	Recall	F-measure	Roc Area	Sensitivity	Specificity	Accuracy (%)
Random Forest	0.955	0.955	0.954	0.988	0.9529	0.9572	95.4575
ANN	0.924	0.924	0.924	0.958	0.9360	0.9085	92.4147
Logistic	0.924	0.924	0.924	0.971	0.9277	0.9182	92.4147
SVM	0.919	0.919	0.919	0.91	0.9182	0.9200	91.8931
Random Tree	0.916	0.916	0.916	0.922	0.9313	0.8921	91.5888
KNN	0.908	0.908	0.908	0.908	0.9212	0.8869	90.7846
Decision Table	0.904	0.903	0.902	0.948	0.8924	0.9226	90.3065
Bayes Net	0.898	0.898	0.898	0.965	0.8969	0.9	89.8066
Naïve Bayes	0.899	0.899	0.898	0.964	0.8973	0.9006	89.8500
RBF	0.845	0.826	0.828	0.9	0.9230	0.7250	82.6125

$$FP\ rate = \frac{FP}{FP + TN} \quad (13)$$

Seven measurements are used to evaluate the performance of the classifiers. Those measures are the precision, recall, F-measure, roc-area, sensitivity, specificity, and accuracy. The results are shown in Table 4 and Figure 2.

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

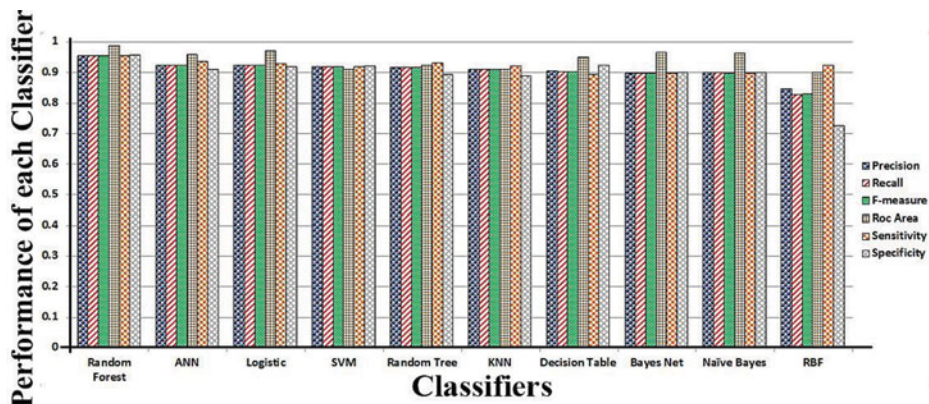
**Figure 2.** The visualization of the classifiers performance on Spambase UCI dataset.

Table 5. A comparison between our proposed method and the previous works.

Author	Dataset	Approach	Accuracy
Supriya & Rahul, 2014	Spambase UCI 4601 Instance	Discretization filter + a set of classifiers (highest classifier accuracy using logistic)	94.45%
Foqaha & Awad, 2016	10-fold cross validation Spambase UCI	Radial Basis Function Particle Swarm Optimization (RBFPSO)	93.1%
Scholar, 2010	Spambase UCI	Multi-layered Perceptron (MLP), and J48	93% for J48, slightly exceeding MLP's 92%,
Chan et al., 2010	Spambase UCI 4601 Instance, 10-fold cross validation	Best stepwise feature selection with a classifier of Euclidean nearest neighbor	82.31% KNN 60.6% Zero rule
Sharma & Arora, 2013	10-fold cross validation algorithm	24 classifiers from the WEKA toolset	Random committee achieved the best result with 94.28% accuracy.
(Saab, Mitri & Awad, 2014)	Spambase UCI 4601 Instance, 10-fold cross validation	SVM, LM-SVM, ANN, and DT	93.4% for SVM, 90.15% for LM-SVM, 92.08% for DT 94.02% for ANN
Our Proposed Method	Spambase UCI 4601 Instance, 10-fold cross validation	Normalization + Infinite latent feature selection + 10 Classifiers applied in WEKA and highest accuracy achieved using Radom Forest	95.45% for Random Forest

Note. RBFPSO = radial basis function particle swarm optimization; MLP = multi-layer perceptron; SVM = support vector machine; DT = decision tree; LM-SVM = local mixture support vector machine; ILFS = infinite latent feature selection.

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$F\text{-measure} = \frac{2TP}{2TP + FP + FN} \quad (16)$$

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

$$Specificity = \frac{TN}{TN + FP} \quad (18)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

It is shown that RF has the highest accuracy of 95.45%, and the other classifiers are sorted according to the accuracy. The experiments were carried out on the platform of core i7 with 3 GHz main frequency and 6G memory, running under window8-64-bit operating system. Our Classification algorithms were used from the WEKA software. Table 5 shows a comparison with the previous work. This comparison is

based on the *spambase* UCI; the database used in our study. Most of the comparison is based on 10-fold cross validation with different classifiers. Supriya and Rahul (2014) worked on the *spambase* dataset and used the approach of discretization filter for preprocessing. They also used a lot of classifiers from the WEKA software to test their system, and the logistic classifier had the highest accuracy of 94.45%. (Foqaha & Awad, 2016) worked on the *spambase* data using Radial Basis Function Particle Swarm Optimization as classifier to identify the ham and spam, achieving an accuracy of 93.1%. M. Scholar (2010) proposed two classifiers for identifying the *spambase*. Those classifiers are multilayer perceptron and J48, suggesting that J48 has a higher performance than multilayer perceptron. Chan, Ji, and Zhao (2010) used best stepwise as a feature selection method and two classifiers. They used KNN based on Euclidian distance and zero rule for classification, achieving 82.31% for KNN and 60.6% for zero rule. Sharma and Arora (2013) performed 10-fold cross validation using 24 classifiers from the WEKA and the random committee, and achieving a high accuracy of 94.28%. Saab, Mitri, and Awad (2014) used SVM, local mixture support vector machine (LM-SVM), ANN, and DT for classifying *spambase* dataset. Their results are 93.4%, 90.15%, 92.08%, and 94.02% respectively. Our proposed method is based on normalization of the *spambase* dataset and infinite latent feature selection. Ten-fold cross validation is applied on data in order not have a lucky split of training and testing data. 10 classifiers were used, achieving 95.45% with the use of random forest.

Conclusion

In this work, a survey about the most efficient techniques about spam and ham is carried out. The survey shows different datasets used for classification. *Spambase* dataset UCI and Enron are the most commonly used datasets in classification. Most of the applications used filtering and machine learning techniques, and some combined them to achieve high performance. Our method achieved highest accuracy using an efficient method for spam e-mails classification on *spam base* UCI datasets. This method includes preprocessing, ILFS for feature selection and data classification using 10 Classifiers. Those classifiers are based on RF, ANN, Logistic Regression, SVM, Random Tree, KNN, Decision Table, Bayes Net, NB, and RBF. The accuracies are 95.4, 92.4, 92.4, 91.8, 91.5, 90.7, 90.3, 89.8, 89.8, and 82.6, respectively. The best performance is realized by using *Random Forest* technique, achieving an accuracy of 95.45%.

Acknowledgment

The authors would like to thank the EELU Students (A. Zaki, A. Husien, O. Mostafa, M. Anwar, A. Ewis, S. Ahmed, G. Naieem) for helping in collecting the *spambase* data required to apply machine learning techniques on it.

References

- Alsmadi, I., & Alhami, I. (2015). Clustering and classification of email contents. *Journal of King Saud University-Computer and Information Sciences*, 27(1), 46–57.
- Araújo, T., Nunes, N., Gamboa, H., & Fred, A. (2015). Generic biometry algorithm based on signal morphology information: Application in the electrocardiogram signal. In *Pattern Recognition Applications and Methods* (pp. 301–310). Cham: Springer.
- Awad, W. A., & ELseuofi, S. M. (2011). Machine Learning methods for E-mail Classification. *International Journal of Computer Applications*, 16(1), 39–45.
- Bhat, S. Y., Abulaish, M., & Mirza, A. A. (2014, August). Spammer classification using ensemble methods over structural social network features. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02* (pp. 454–458). Washington, DC, USA: IEEE Computer Society.
- Chan, T. Y., Ji, J., & Zhao, Q. (2010). Learning to Detect Spam: Naive-Euclidean Approach. *International Journal of Signal Processing*, 1, 31–38.
- Chen, X. L., Liu, P. Y., Zhu, Z. F., & Qiu, Y. (2009, August). A method of spam filtering based on weighted support vector machines. In *IT in Medicine & Education, 2009. ITIME'09. IEEE International Symposium on* (Vol. 1, pp. 947–950). Jinan, China: IEEE.
- Christina, V., Karpagavalli, S., & Suganya, G. (2010). A study on email spam filtering techniques. *International Journal of Computer Applications*, 12(1), 7–9.
- Foqaha, M. A. M. (2016). Email Spam Classification Using Hybrid Approach of RBF Neural Network And Particle Swarm Optimization. *International Journal of Network Security & Its Applications*, 8(4), 17–28.
- Gomez, J. C., & Moens, M. F. (2012). PCA document reconstruction for email classification. *Computational Statistics & Data Analysis*, 56(3), 741–751.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1), 10–18. Retrieved from <https://sourceforge.net/projects/weka/last> accessed on 8-3-2018.
- Haykin, S. (2008). *Neural networks and learning machines* (3rd ed.). New Jersey: Pearson Prentice Hall.
- Kaur, G., & Oberai, E. N. (2014). A review article on Naive Bayes classifier with various smoothing techniques. *International Journal of Computer Science and Mobile Computing*, 3(10), 864–868.
- Kiran, P., & Atmosukarto, I. Spam or Not Spam-That is the question. Tech. rep., University of Washington. <http://www.cs.washington.edu/homes/indria/research/spamfilter/ravi.indri.pdf>. last accessed on 5-6-2018.
- Klimt, B., & Yang, Y. (2004, September). The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning* (pp. 217–226). Berlin, Heidelberg: Springer. Enron dataset. <https://www.cs.cmu.edu/~enron/> last accessed on 8-3-2018.
- Kohavi, R. (1995, April). The power of decision tables. In *European conference on machine learning* (pp. 174–189). Berlin, Heidelberg: Springer.
- Kumar, R. K., Poonkuzhali, G., & Sudhakar, P. (2012, March). Comparative study on email spam classifier using data mining techniques. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1, pp. 14–16). Hong Kong.
- Manisha, A. S., & Jain. (2015). M. D. R. Data pre-processing in spam detection. *International Journal of Science Technology & Engineering*, 1(1), 33–37.
- Mohd, A., Yuk, Y., Wei-Chang, Y., Noorhaniza, W., & Ahmad, M. (2011). Classification technique using modified particle swarm optimization. *Modern Applied Science*, 5(5), 150–164.
- Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, & A. J. Smola, (Eds.), *Advances in Kernel Methods* (pp. 185–208). Cambridge, MA: MIT Press.

- Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, 9(2), 181–199.
- Provost, J. (1999). Naive-bayes vs. rule-learning in classification of email. *Technical Report AITR-99-284, University of Texas at Austin, Artificial Intelligence Lab*.
- Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. (22), pp. 41–46). Sicily, Italy: IBM.
- Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 93–104.
- Roffo, G., Melzi, S., Castellani, U., & Vinciarelli, A. (2017). Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1398–1406). Venice.
- Roy, A., Govil, S., & Miranda, R. (1995). An algorithm to generate radial basis function (RBF)-like nets for classification problems. *Neural networks*, 8(2), 179–201.
- Saab, S. A., Mitri, N., & Awad, M. (2014, April). Ham or spam? A comparative study for some content-based classification algorithms for email filtering. In *Electrotechnical Conference (MELECON), 2014 17th IEEE Mediterranean* (pp. 339–343). IEEE.
- Scholar, M. (2010). Supervised learning approach for spam classification analysis using data mining tools. *Organization*, 2(8), 2760–2766.
- Sharaff, A., Nagwani, N. K., & Dhadse, A. (2016). Comparative study of classification algorithms for spam email detection. In *Emerging research in computing, information, communication and applications* (pp. 237–244). New Delhi: Springer.
- Sharma, S., & Arora, A. (2013). Adaptive approach for spam detection. *International Journal of Computer Science Issues*, 10(4), 23–26.
- Silva, R. M., Yamakami, A., & Almeida, T. A. (2012, December). An analysis of machine learning methods for spam host detection. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on* (Vol. 2, pp. 227–232). Boca Raton, FL, USA: IEEE.
- Spambase dataset. <https://archive.ics.uci.edu/ml/datasets/spambase>. Hopkins, M., Reeber, E., Forman, G., Suermondt, J., creators, Hewlett-Packard Labs.
- Supriya, S., & Rahul, P. (2014, November). Improving spam mail filtering using classification algorithms with discretization filter. *International Journal of Emerging Technologies in Computational and Applied Sciences*, 10(1), 82–87.
- Trivedi, S. K., & Dey, S. (2013). Interplay between probabilistic classifiers and boosting algorithms for detecting complex unsolicited emails. *Journal of Advances in Computer Networks*, 1(2), 132–136.
- Vapnik, V. (2013). *The nature of statistical learning theory*. Verlag, New York: Springer Science & Business Media.
- Yan, J., & Lee, J. (2005). Degradation assessment and fault modes classification using logistic regression. *Journal of manufacturing Science and Engineering*, 127(4), 912–914.
- Youn, S., & McLeod, D. (2007). A comparative study for email classification. In *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, (pp. 387–391). Dordrecht: Springer.
- Yu, B., & Xu, Z. B. (2008). A comparative study for content-based dynamic spam classification using four machine learning algorithms. *Knowledge-Based Systems*, 21(4), 355–362.
- Zhang, L., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4), 243–269.