Highly Accurate Spam Detection with the Help of Feature Selection and Data Transformation

Hidayet Takci
Computer Engineering Department, Sivas Cumhuriyet University, Turkey
htakci@cumhuriyet.edu.tr

Fatema Nusrat
Computer Science Department, Asian University for Women, Bangladesh
fatema.nusrat@auw.edu.bd

Abstract: *The amount of spam is increasing rapidly while the popularity of emails is increasing. This situation has led to the need to filter spam emails. To date, many knowledge-based, learning-based, and clustering-based methods have been developed for filtering spam emails. In this study, machine-learning-based spam detection was targeted, and C4.5, ID3, RndTree, C-Support Vector Classification (C-SVC), and Naïve Bayes algorithms were used for email spam detection. In addition, feature selection and data transformation methods were used to increase spam detection success. Experiments were performed on the UC Irvine Machine Learning Repository (UCI) spambase dataset, and the results were compared for accuracy, Receiver Operating Characteristic (ROC) analysis, and classification speed. According to the accuracy comparison, the C-SVC algorithm gave the highest accuracy with 93.13%, followed by the RndTree algorithm. According to the ROC analysis, the RndTree algorithm gave the best Area Under Curve (AUC) value of 0.999, while the C4.5 algorithm gave the second-best result. The most successful methods in terms of classification speed are Naïve Bayes and RndTree algorithms. In the experiments, it was seen that feature selection and data transformation methods increased spam detection success. The binary transformation that increased the classification success the most and the feature selection method was forward selection.*

Keywords: *Internet security, prediction methods, feature selection, data conversion, spam detection.*

Received July 2, 2021; accepted September 28, 2022
https://doi.org/10.34028/iajit/20/1/4

1. Introduction

Email is one of the most frequently used services on the internet, as it is a low-cost, popular and fast communication tool [41]. However, the increasing number of spam emails makes email communication problematic and insecure [12, 25]. Spam mails are a major security concern as they contain misleading banking transactions, phishing, and malware attacks [19]. They host many viruses, trojans, phishing attacks, and malware [29]. In addition to these problems, spam e-mails consume network traffic in vain, making it difficult to deal with them [3].

The first thing to be done to struggle with spam emails is to classify emails according to whether they are spam or not [34]. The changing content of spam emails makes it difficult to classify them, but the classification is necessary to fight spam [4]. Many knowledge-based and machine learning-based methods have been proposed so far for the classification of spam emails [15, 23, 35]. In particular, the number of methods based on machine learning is increasing day by day. In this context, probability-based, decision tree-based, support vector machines-based [9], artificial neural networks-based [11], and state-based [14] studies are used in spam detection.

One of the algorithms frequently used in spam detection is the Naive Bayes algorithm [5]. In a study by Yitagesu and Tijare [42], the naïve bayes algorithm was used together with Support Vector Machines (SVM) and the K-Nearest Neighbor algorithm (K-NN) for spam detection. In that study, the Naïve Bayes algorithm gave the highest classification accuracy. Sao and Prashanti [33] used Naive Bayes and Support Vector Machine algorithms together and experimented with the Lingspam dataset. In that study, the Naïve Bayes algorithm outperformed the support vector machine algorithm. Rusland *et al*. [32] used the Naïve Bayes algorithm on two datasets and compared the results according to accuracy, recall, precision, and F-measure. The study was handled by three stages data preprocessing, feature selection, and classification with Naive Bayes. In experiments with the WEKA tool, 91.13% accuracy on Spam Data and 82.54% accuracy on the SpamBase dataset were obtained.

Decision tree-based methods are also frequently used in spam detection. In one of these studies, Balakumar and Ganeshkumar [7] used decision tree algorithms named J48, Rndtree, BFtree, REPtree, Logistic Model Tree (LMT), and simple Classification and Regression Tree (CART) for spam detection. Classifiers were evaluated on the UC Irvine Machine Learning Repository (UCI) spambase dataset and with the Weka tool. According to the classification results, the most successful classifier was the RndTree algorithm with 99%. In the study by Shrivastava and Dubey [37], Naive Bayes, RandomForest, RandomTree, REPTree and J48 algorithms were run on the UCI spambase dataset and

92% classification accuracy was obtained. Sharaf *et al*. [36] used decision tree algorithms named ID3, J48, Simple Cart, and Active Directory Tree (AD Tree) for spam detection. The dataset used is Enron, the machine learning tool used is Weka. The highest accuracy rate obtained in the study belongs to the J48 algorithm with 92.7%. In another study, Bassiouni *et al*. [8] used ten different classification algorithms for spam detection. The dataset used in the study is the UCI spambase dataset. The highest classification accuracy they obtained belongs to the Random Forest algorithm with 95.45%. Spam detection has been made not only for e-mails but also for Twitter messages [28]. The data set used in the study on Twitter messages was obtained with the Twitter API. At the end of the experimental studies, the Random Forest algorithm gave the best results. The f-measure value obtained was 95.7%.

Awad and Foqoha [6] used a combination of Radial Basis Function Neural Network (RBFNN) and Particle Swarm Optimization (PSO) algorithms for spam detection. The highest classification accuracy in experiments, with different numbers of layers and different numbers of nodes, was obtained as 93.1%. A method based on natural language processing has been used for spam detection [13] and more effective filtering of spam emails is provided by word root finding and hashing technique. Abdulhamid *et al*. [1] presented an approach based on performance analysis using some classification techniques such as bayesian logistic regression, hidden naïve bayes, logit boost, rotation forest, logistic model tree, and REP Tree. Techniques were compared with accuracy, precision, recall, f-measure, mean square error, receiver operator characteristic area, and root relative squared error using the spambase dataset and WEKA data mining tool. By the experimental results, the highest accuracy and the lowest accuracy were 0.942, and 0.891 according to the Rotation Forest algorithm and the REP Tree algorithm. Yüksel *et al*. [43] used the Support Vector Machine and Decision tree for spam filtering. Algorithms are trained and evaluated through the Microsoft Azure platform. The result of the SVM method is 97.6% and for the Decision tree is 82.6%. The result indicates that the SVM classifier outperforms Decision Tree (DT).

It has emerged as a solution method in using natural language processing approaches to detect spam. Kontsewayaa *et al*. [22] used Naive Bayes, K Nearest Neighbor, SVM, Logistic regression, Decision tree, and Random forest algorithms for spam detection supported by natural language processing. According to the results of this study, Logistic regression and NB gave 99% accuracy. The results showed that filtering methods can be used to create a smarter spam detection classifier. Srinivasan *et al*. [38] presented the effect of word embedding in deep learning for email spam detection, the proposed method outperformed other classical email representation methods. The rapid development in the field of Internet of Things leads to many malicious attacks as it hosts many smart objects that do not have an effective security framework. In a study by Manoharan *et al*. [26] is proposed for multi-channel Convolutional Neural Network (CNN) malware detection. This model has two channels connected in parallel, where one CNN receives an opcode sequence as input and the other CNN operates with system calls. Performance analyzes for the aforementioned study were performed and evaluated using accuracy, precision, recall, F1-measure and time. Experimental results show that multi-channel CNN outperforms other considered techniques, achieving a high accuracy of 99.8% to classify malicious samples from benign ones. Word embeddings were used in spam detection by AbdulNabi and Yaseen [2] and classification was made with the Bidirectional Encoder Representations from Transformers (BERT) model. The results obtained from the experiments were compared with baseline DNN, classical KNN, and Naïve Bayes algorithms. The highest classification success achieved with the BERT model was reported as 98.67%. In addition to natural language processing methods, optimization methods are also used in spam detection. In one of these studies, Mashaleha *et al*. [27] integrated the k nearest neighbor algorithm with the Harris Hawk Optimizer algorithm and used it in spam detection. The spam detection success of the proposed model surpassed other optimization algorithms and 94.3% accuracy was achieved.

In this study, C4.5, RndTree, Naive Bayes, C-SVC, and ID3 algorithms are used for spam detection. Feature selection and data transformation are used to increase the success of spam detection. In feature selection, Backward Logit, Forward Logit, and ReliefF algorithms are applied to the original feature set. Then the effect of the selected features on the classification success was measured. Relative transformation, binary transformation, and logarithmic transformation are performed at the data transformation stage, and the results are reported. The output of the study is to show the effect of feature selection and data transformation after finding the most effective classification algorithm in UCI spambase data.

2. Methodology

This study aims to introduce a machine learning-based model that will filter spam emails. With the help of machine learning algorithms, spam detection models will be created from the training data then spam detection will be made on the test data. In our study, C4.5, ID3, RndTree, C-SVC, and Naive Bayes algorithms are used. Feature selection and data transformation methods will be applied to the data to increase spam detection success. For feature selection, forward selection, fisher filtering, and relief methods are used for data transformation, relative, binary and logarithmic transformation.

2.1. Classification Algorithms

Three of the algorithms to be used in spam detection are decision tree-based algorithms. Others are support vector machine-based C-SVC algorithm and Naïve Bayes algorithm. The selected algorithms are classifiers with high performance and previously used in spam detection.

ID3 and C4.5 are machine learning algorithms introduced by Quinlan to obtain a decision tree from a dataset [30]. The ID3 algorithm is the predecessor of the C4.5 algorithms. Both algorithms construct a tree from the labeled training data with the help of information gain. For the information gain calculation, the entropy values of the parent and child nodes are needed. The homogeneity of the parent node, t being the parent node, is calculated as in Equation (1) below.

$$Entropy(t) = -\sum_{j} p(j \mid t) \log\, p(j \mid t) \tag{1}$$

The entropy value is calculated for pre-partition (parent) and post-splitting (i), and information gain is obtained as seen in Equation (2).

$$GAIN = Entropy(parent) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right) \tag{2}$$

The Entropy value is obtained before and after the splitting of the parent node, and information gain is calculated from the difference. If the information gain is a positive value, the node is split otherwise it is not. Thus, the tree expands or stops. Since decision trees are easy to interpret and develop, they have been used to solve many problems until today.

Random tree, which is a category of decision trees, creates a type of random forest algorithm [10]. Thanks to the Ensemble technique, classification performance is improved with the help of re-weighting and training data.

Support vector machines are one of the most powerful classifiers in machine learning. The equations for the linear support vector machine are as given below. The data points x, and w denotes the coefficients, and b is the intersection value.

$$\begin{aligned} w^T x+b &= 0 \\ w^T x+b &< 0 \\ w^T x+b &> 0 \end{aligned} \tag{3}$$

The purpose of the algorithm is to separate the classes from each other in a way that maximizes the margin. In margin maximization, support vectors are taken into account instead of all data. The distance between the support vectors and the hyperplane is 1/||w|| and the distance between the support vectors of both classes is 2/||w||. By maximizing the margin, points in different classes will be distributed as far as possible from each other then a more successful classification will be made.

The algorithm was originally designed for linearly separable spaces and later became usable for nonlinear forms. The four basic kernel functions are seen in Table 1.

Table 1. Kernel functions.

Linear kernel	$K(x, y)=x^T y$
Polynomial kernel (degree of d)	$K(\mathbf{x}, \mathbf{y}) =(x^T y+c)^d$
Radial basis kernel (RBF)	$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-y\|^2}{2\sigma^2}}$
Sigmoid kernel	$K(\mathbf{x}, \mathbf{y}) =\tanh(\alpha x^T y+c)$

Linear kernel and RBF kernel are often used from kernel functions. In this study, a support vector classifier with a linear kernel named C-Support Vector Classifier (C-SVC) was used from the LibSVM library [40].

The naïve bayes algorithm is a fast algorithm that makes classification based on probability theory [24]. According to Bayes' theorem, the probability of event A to be in class C can be given as follows, to present a single event A.

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)} \tag{4}$$

When event *A* is presented with more than one attribute value, the probability of it occurring in class *Cj* was given event A will be presented as follows.

$$P(C_j \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C_j)P(C_j)}{P(A_1 A_2 \ldots A_n)} \tag{5}$$

To solve the above equation, $P(A_1 A_2 \ldots A_n \mid C_j)$ value needs to be obtained. This value is calculated according to the naive approach as follows.

$$P(A_1, A_2, \ldots, A_n \mid C_j) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j) \tag{6}$$

2.2. Feature Selection

In this study, Backward Elimination, Forward Selection, ReliefF, and Fisher Filtering methods were used for feature selection. Backward Elimination and Forward Selection methods are Stepwise regression models. Stepwise regression is a method of fitting regression models in which the selection of prediction variables is performed by an automatic procedure.

The Backward elimination method starts with all candidate variables and tests variable removal according to a selected criterion and confirms variable deletion when variable deletion causes an insignificant change in the model [18]. This process continues until no candidate variable remains. Methods such as F-test or t-test are used as selection criteria. Forward selection starts with a zero variable and tests the addition of a candidate variable to the feature set according to a selected criterion, then aims to add the candidate variable even if adding a variable causes a significant improvement.

ReliefF algorithm is a feature selection method that was developed by Kira and Rendell [20] and uses the filtering technique in feature selection. It was first developed by binary classification problems with discrete and numerical properties. Then, a score-based method was added for feature selection. Thus, a ranking

is obtained from important features to unimportant features according to the score order.

One of the methods used in this study is the Fisher Filtering method. It is one of the supervised learning techniques using the filtering method in feature selection [16]. The Fisher filtering method obtains a score for each candidate variable. The purpose of this score is to obtain features with high distinctiveness. Characteristics with high scores are those with a high correlation with the target variable.

2.3. Data Transformation

The success of machine learning algorithms is affected by the transformation of data. While the data with reduced variance sometimes affects the higher classification success, sometimes it can be the opposite. In order to present *fr*, an e-mail, fr_i, the i[th] attribute value in that mail, the conversion from fr_i to $fr_i^{new}$was performed by three different methods:

1) Relative transformation.
2) Binary transformation.
3) Logarithmic transformation.

Thanks to the relative transformation, property values are scaled to the range 0-1. Transformation, it can be presented as $fr_i^{new} = fr_i / \sum_{i=1}^{m} fr_i$. Relative transformation allows equal representation of vectors of different lengths. Binary transformation refers to the conversion of continuous data to binary data. The operation is performed according to the following transformation.

if fr_i>0

$$fr_i^{new} = 1$$

else

$$fr_i^{new} = 0$$

In logarithmic transformation, logarithm values are taken according to the base 10 of the feature values. The conversion is presented as $fr_i^{new} = log_{10}(fr_i)$. By using this transformation, the effect of deviations in the data is minimized.

3. Experimental Study

In this study, the UCI spambase dataset [39] was used to evaluate the performance of spam detection. Spambase dataset contains 4601 records that consist of 58 attributes. Details of these attributes are presented in Table 2.

Table 2. UCI spambase dataset details

Attributes	Descriptions	Type
w1-w48	Number of times the word appeared in the email/total number of words in the email	Continuous
w49-w54	Total number of some characters/total number of characters in email	Continuous
w55	The average length of continuous capital letter strings	Continuous
w56	Length of a longest continuous string of capital letters	Continuous
w57	Total capitalization in email	Integer
w58	Indicates whether the e-mail is considered spam (1) (0), that is, whether it is considered spam.	{0,1}

Each attribute is labeled as w1, w2…, w57 for ease of tracking. The last attribute (w58) introduces the class variable.

Spambase data was classified according to the algorithms specified in the method section. The classification was made with the original data set then the data with feature selection was applied. Then, the results were obtained by applying data transformation to these data. Thus, after the classifier performance in the original data was determined, the performance after feature selection and data transformation was obtained. A comparison of methods was performed based on accuracy and ROC analysis. The preferred model evaluation method for comparing algorithms is the 10-fold cross-validation method. After comparing the methods in terms of accuracy, it was compared in terms of processing time and previous studies. The experiments were carried out with the help of a machine learning tool called Tanagra [31]. The parameters used in the experiments are given in Table 3.

Table 3. Algorithms and parameter values.

Algorithms	The parameters and values	
C4.5	The minimum size of leaves	5
	Confidence level	0.25
ID3	Min size for split	200
	Min size of leaves	50
	Max depth of the tree	10
RndTree	Selected attributes	-1
C-SVC	The degree of kernel function	1
	Gamma	0
	Coefficient 0	0
	The complexity	1
Naïve Bayes	Lambda	0

Firstly, classifiers were run with original values (non-selected and non-transformed). Then, classifiers are run with selected features and transformed values. Finally, the results are compared. Thus the effect of feature selection and feature transformation in the classifiers can be seen.

3.1. Results based on Feature Selection and Feature Transformation

At the beginning of the experimental studies, feature selection algorithms were run for sub-feature sets. The sub-feature sets obtained from the feature selection algorithms are given in Table 4.

Table 4. Feature selection methods and sub-feature sets.

Feature selection	Sub feature sets
Forward selection	w57, w7, w23, w53, w25, w27, w17, w16, w12, w46, w2, w45, w42, w38, w56, w39, w52, w55, w30, w33, w49
Fisher Filtering	w57, w56, w55, w7, w27, w23, w25, w53, w30, w26
ReliefF	w27, w32, w2, w34, w40, w57, w28, w19, w12, w30

According to Fisher filtering and ReliefF method, the best 10 features with high classification were selected. On the other hand, 21 features were selected

with the Forward Selection method. The backward elimination method, on the other hand, did not select the feature set. For this reason, the output of backward elimination could not be used in the experiments.

In addition to feature selection, the effect of data transformation was also evaluated in this study. For this, results were found according to three different transformation methods, the details of which are presented in the methodology section. Therefore, five different algorithms were run according to a total of 16 different setups for four different feature sets (one with all features and the other three with the output of feature selection methods) and four different data values (one original data and three transformed data) were then compared in terms of ROC analysis. Summary information about 80 experiments performed is presented in Tables 5 and 6.

Table 5. Comparison of algorithms in terms of correct recognition.

Feature selection	Data transformation	C4.5	ID3	Rnd Tree	C-SVC	Naïve Bayes
All attributes	Original	90.08	88.76	90.83	80.89	78.87
	Relative transformation	90.78	88.76	90.59	80.09	78.87
	Binary transformation	91.28	88.80	90.91	93.13	86.76
	Logarithmic transformation	89.85	86.22	90.07	83.85	75.43
Forward selection	Original	91.33	88.33	90.43	78.76	77.07
	Relative transformation	91.33	88.33	90.80	78.70	77.11
	Binary transformation	90.07	88.87	88.76	90.70	88.57
	Logarithmic transformation	91.13	87.54	89.98	80.02	75.83
Fisher Filtering	Original	88.50	85.87	86.80	73.70	74.89
	Relative transformation	88.50	85.87	86.80	73.70	74.89
	Binary transformation	87.72	86.96	86.89	87.35	87.37
	Logarithmic transformation	89.24	84.61	87.96	78.43	76.02
ReliefF	Original	80.33	77.46	78.85	69.67	68.00
	Relative transformation	80.33	77.46	78.85	69.67	68.00
	Binary transformation	73.85	70.46	72.93	68.20	74.91
	Logarithmic transformation	81.54	77.04	80.13	76.33	73.96

Table 6. Comparison of classifiers in terms of ROC analysis.

Feature selection	Data transformation	C4.5	ID3	Rnd Tree	C-SVC	Naïve Bayes
All attributes	Original	0.9901	0.9550	0.9988	0.8872	0.9042
	Relative transformation	0.9859	0.9550	0.9989	0.8872	0.8281
	Binary transformation	0.9810	0.9500	0.9903	0.8872	0.8990
	Logarithmic transformation	0.9797	0.9270	0.9989	0.8872	0.8270
Forward selection	Original	0.9860	0.9530	0.9980	0.8870	0.7810
	Relative transformation	0.9860	0.9530	0.9980	0.8870	0.7810
	Binary transformation	0.9620	0.9460	0.9701	0.8870	0.9110
	Logarithmic transformation	0.9810	0.9330	0.9990	0.8870	0.8260
Fisher Filtering	Original	0.9770	0.9560	0.9900	0.8870	0.7310
	Relative transformation	0.9770	0.9560	0.9900	0.8870	0.7310
	Binary transformation	0.9390	0.9490	0.9470	0.8870	0.9380
	Logarithmic transformation	0.9780	0.9150	0.9890	0.8870	0.8340
ReliefF	Original	0.9380	0.8560	0.9540	0.8870	0.6560
	Relative transformation	0.9380	0.8560	0.9540	0.8870	0.6560
	Binary transformation	0.8520	0.8510	0.8550	0.8870	0.7930
	Logarithmic transformation	0.9470	0.8700	0.9780	0.8870	0.8190

Table 5 shows the effect of feature selection and data transformation methods on classifiers. The classifier C-SVC algorithm gives the highest accuracy obtained as a result of the studies done. C-SVC algorithm was followed with RndTree and C4.5 algorithms. This result was obtained with the help of binary transformation on the original data set. Naïve Bayes was the algorithm that gave the lowest result in experimental studies. This value was obtained in the data set in which the feature set obtained with the ReliefF algorithm was subjected to a relative transformation.

Accuracy-based comparison is not an adequate comparison method on its own, especially in unbalanced data sets. Therefore, ROC analysis based on false positive rate and true positive rate values was needed. ROC analysis helps us to choose the model that is successful in binary classification problems. In this study, the performance of each model is given by the area under the curve value. Again, results were obtained for 16 different setups and 5 different algorithms, and the results are presented in Table 6.

When we compared the classifiers according to the AUC values, the RndTree algorithm gave the highest performance. It was followed with C4.5 and ID3 algorithms, respectively. The C-SVC algorithm, which gave the best results by the accuracy rates, gave one of the most unsuccessful results according to the ROC values. As with the accuracy rates, the Naïve Bayes algorithm gave the lowest result. In addition, the AUC values obtained from the C-SVC algorithm gave almost the same results in all experiments. In terms of a feature selection effect, original features and forward selection

outputs gave similar results. In data transformation, the original features and log transformation gave better results. The most successful transformation method was the relative transformation method. The feature selection method that gave the lowest result was ReliefF.

3.2. Comparison of Algorithms in Terms of Processing Times

Spam detection is one of the applications where classification speed is needed. Therefore, the measurement of classification speed will be an option in classifier selection. Processing times were measured when comparing the classifiers in terms of speed. The values were measured on a computer with an Intel (R) Core (TM) i5-6200U processor and 16 GB of physical memory. The measurement of processing times is presented below. In addition, since the processing times are related to the number of features, the effect of the feature selection method in the time measurement was also measured. So, it will be possible to choose the lower dimensional one out of two feature sets that give the same result. The relationship between feature conversion and duration was not measured. Because the duration that affects the number of features is not related to the value of the feature. Algorithms in terms of processing time are presented in Table 7.

Table 7. Algorithms in terms of processing time.

	Time (ms)			
Algorithm	**Orjinal**	**Forward Selection**	**Fisher Filtering**	**ReliefF**
C4.5	5109	2390	1360	1360
ID3	2313	1562	734	2254
RndTree	1968	1234	937	890
C-SVC	15407	13422	8485	8954
Naive Bayes	546	375	375	359

The fastest algorithm according to the classification times is the naïve bayes algorithm. This result explains why the Naïve Bayes algorithm is used in commercial products. The method with the longest classification time is the C-SVC algorithm. The complexity of the algorithm affected the classification speed. In addition, feature selection methods also affected the duration in general. Datasets with fewer features were classified in a shorter time, while others took longer to classify. The RndTree algorithm, which is at the top in accuracy and ROC values, was the second most successful algorithm in terms of time. Although the Naïve Bayes algorithm is the fastest, its low performance in terms of the other two factors affects the algorithm negatively.

3.3. Comparison with Past Studies

In this section, some studies use machine learning algorithms for spam detection. Our study was compared with the studies in the literature. Summary information is given in Table 8.

Table 8. Comparison of our study and other studies.

Author(s)	Method	Dataset	Accuracy/Performance
Rusland *et al*. [32]	Naïve Bayes	Spam data UCI Spambase dataset	91.13% 82.54%
Balakumar and Ganeshkumar [7]	J48, Rndtree, BFtree, REPtree, LMT and simple CART	UCI Spambase dataset	RndTree (99%)
Shrivastava and Dubey [37]	Naive Bayes, RandomForest, RandomTree, REPTree and J48	UCI Spambase dataset	RandomTree (About 92%)
Sharaf *et al*. [36]	ID3, J48, Simple Cart and ADTree	Enron dataset	J48 (92.7%) ID3 (89.1%) ADTree (90.9%) Simple Cart (92.6%)
Bassiouni *et al*. [8]	Random Forest	UCI Spambase data	Random Forest (95.45%)
McCord and Chuah [28]	Random Forest	Twitter dataset	f-score = 95.7
Awad and Foqoha [6]	RBFNN ve PSO	UCI Spambase data	RBFNN & PSO (93.1%)
Abdulhamit *et al*. [1]	Bayesian Logistic Regression, Hidden Naïve Bayes, Logit Boost, Rotation Forest, Logistic Model Tree, REP Tree	UCI Spambase dataset	Rotation Forest (94.2%)
Our study	C4.5, ID3, RndTree, C-SVC, Naive Bayes	UCI Spambase dataset	C-SVC (93.13%)

Knowledge-based and machine learning-based methods are frequently used in spam detection. Machine learning-based methods are also known as content filtering methods. Naïve Bayes, Random Forest, Support Vector Machines, Decision Tree Algorithms, and artificial neural networks have been used in machine learning-based spam detection. Similar to the literature, Decision Trees, Support Vector Machine, and Naïve Bayes algorithm were used for spam detection in our study. In spam detection studies, data sets such as Spambase [39], Enron [21], and Spam assassin [17] were preferred. The most frequently used dataset in our research is the UCI spambase dataset. Therefore, the UCI spambase dataset was used in this study. One of the remarkable issues in past studies is the frequent use of machine

Learning tools such as Weka and Rapidminer in experiments [7, 36, 42]. In this study, another machine learning tool called Tanagra was used.

RndTree and Support Vector Machines have been the best-performing classifiers in past studies. According to the results obtained in our study, the classifiers that gave the best results were C-SVC and RndTree. The accuracy value we obtained is at the level of the literature. In addition, the effect of data transformation on success was seen in our study. The improvement achieved through data transformation was 2.46%.

4. Discussion

A large number of machine learning algorithms have been used to date for spam detection. In this study, five different machine learning algorithms, three of which are decision tree-based, were used for spam detection. Feature selection and data transformation were applied to increase the detection success of the algorithms. Within the scope of feature selection, four different feature selection methods were run. However, the Backward Elimination method did not select in this data set. Therefore, in the experiments, sub-feature sets obtained according to the Forward Selection, Fisher Filtering, and ReliefF methods were used. Relative transformation, binary transformation, and logarithmic transformation were used within the scope of data transformation.

When the studies were compared in terms of accuracy, the most successful algorithm was the C-SVC algorithm. The highest correct recognition rate obtained by the algorithm was measured as 93.13% and was followed by the RndTree algorithm. The most unsuccessful result was given with the Naïve Bayes algorithm. In experiments based on feature selection, the forward selection method gave better results than the others. In data transformation, on the other hand, the success of binary transformation was seen with a clear difference. The correct recognition rate was 80.89% before binary transformation, then after binary transformation reached 93.13%.

The next model comparison method was ROC analysis. The algorithm RndTree gave the best result according to the ROC analysis. The RndTree algorithm was followed by the C4.5 algorithms. The most unsuccessful algorithm was again the Naïve Bayes algorithm. The highest value obtained according to the ROC values was measured as 0.999. Spam detection is an application that needs classification speed. Therefore, the third metric used to compare methods was classification speed. The Naïve Bayes algorithm gave the best result in terms of classification time. It followed the RndTree algorithm. In addition, the effect of decreasing feature numbers due to feature selection on classification speed was observed.

5. Conclusions

As a result of the studies, the most successful classification algorithms were C-SVC and RndTree algorithms. The c-SVC algorithm gave the best result in terms of correct recognition rate, and the RndTree algorithm gave the highest result in terms of ROC analysis. The algorithm that gave the most successful result in terms of time was the Naïve Bayes algorithm. The effect of feature selection and data transformation on classification was also examined, and both feature selection and data transformation had a positive effect on classification. The sub-feature set, which is the output of the forward selection method, gave as successful results as the original feature set in a shorter processing time. The increase in classification accuracy obtained by data transformation was 2.46%. The classification success obtained with single classifiers and data transformation is 93.13%. Support vector machines and decision tree algorithms are suitable algorithms for spam detection. In particular, the RndTree algorithm gave the best results for spam detection.

References

[1] Abdulhamid S., Shuaib M., Osho O., Ismaila I., and Alhassan J., "Comparative Analysis of Classification Algorithms for Email Spam Detection," *International Journal of Computer Network and Information Security*, vol. 10, no. 1, pp. 60-67, 2018.

[2] AbdulNabi I. and Yaseen Q., "Spam Email Detection Using Deep Learning Techniques," *Procedia Computer Science*, vol. 184, pp. 853-858, 2021.

[3] Ablel-Rheem D., Ibrahim A., Kasim S., Almazroi A., and Ismail M., "Hybrid Feature Selection and Ensemble Learning Method for Spam Email Classification," *International Journal*, vol. 9, no. 1.4, pp. 217-223, 2020.

[4] Almeida T., Almeida J., and Yamakami A., "Spam Filtering: How The Dimensionality Reduction Affects The Accuracy of Naive Bayes Classifiers," *Journal of Internet Services and applications*, vol. 1, no. 3, pp. 183-200, 2011.

[5] Androutsopoulos I., Paliouras G., and Michelakis E., "Learning to Filter Unsolicited Commercial E-Mail Technical Report," *Technical Report* 2004/2, NCSR Demokritos, 2006.

[6] Awad M. and Foqaha M., "Email Spam Classification Using Hybrid Approach of RBF Neural Network and Particle Swarm Optimization," *International Journal of Network Security and its Applications*, vol. 8, no. 4, pp. 17-28, 2016.

[7] Balakumar C. and Ganeshkumar D., "A Data Mining Approach on Various Classifiers in Email Spam Filtering," *International Journal for Research in Applied Science and Engineering Technology*, vol. 3, no. 1, pp. 8-14, 2015.

[8] Bassiouni M., Ali M., and El-Dahshan E., "Ham and Spam E-Mails Classification Using Machine Learning Techniques," *Journal of Applied Security Research*, vol. 13, no. 3, pp. 315-331, 2018.

[9] Bouguila N. and Amayri O., "A Discrete Mixture-Based Kernel For Svms: Application to Spam and Image Categorization," *Information Processing and Management*, vol. 45, no. 6, pp. 631-642, 2009.

[10] Breiman L., "RANDOM FORESTS," *Machine Learning*, vol. 45, pp. 5-32, 2001.

[11] Cao Y., Liao X., and Li Y., "An E-Mail Filtering Approach Using Neural Network," *in International Symposium on Neural Networks*, pp. 688-694, 2004.

[12] DeBarr D. and Wechsler H., "Spam Detection Using Random Boost," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1237-1244, 2012.

[13] Dhanaraj K. and Thiag H., "Email Classification for Spam Detection Using Word Stemming," *International Journal of Computer Applications*, vol. 1, no. 5, pp. 45-47, 2010.

[14] Fdez-Riverola F., Iglesias E., Díaz F., Méndez J., and Corchado J., "SpamHunting: An Instance-Based Reasoning System for Spam Labelling and Filtering," *Decision Support Systems*, vol. 43, no. 3, pp. 722-736, 2007.

[15] Feng W., Sun J., Zhang L., Cao C., and Yang Q., "A Support Vector Machine Based Naive Bayes Algorithm for Spam Filtering," *in Proceedings IEEE 35th International Performance Computing and Communications Conference*, Las Vegas, pp. 1-8, 2016.

[16] Gu Q., Li Z., and Han J., "Generalized Fisher Score for Feature Selection," *in Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, Barcelona, pp. 266-273, 2011.

[17] Guzella T. and Caminhas W., "A Review of Machine Learning Approaches to Spam Filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206-10222, 2009.

[18] Halinski R. and Feldt L., "The Selection of Variables in Multiple Regression Analysis," *Journal of Educational Measurement*, vol. 7, no. 3, pp. 151-157, 1970.

[19] Heron S., "Technologies for Spam Detection," *Network Security*, vol. 2009, no. 1, pp. 11-15, 2009.

[20] Kira K. and Rendell L.,"The Feature Selection Problem: Traditional Methods and A New Algorithm," *Aaai*, vol. 2, no. 1992a, pp. 129-134, 1992.

[21] Klimt B. and Yang Y., "Introducing the Enron Corpus," *CEAS*, 2004.

[22] Kontsewayaa Y., Antonova E., Artamonovb A., "Evaluating the Effectiveness of Machine Learning Methods for Spam Detection," *Procedia Computer Science*, vol. 190, pp. 479-486, 2021.

[23] Kumar S., Gao X., Welch I., and Mansoori M., "A Machine Learning Based Web Spam Filtering Approach," *in Proceedings of the IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, Crans-Montana, pp. 973-980, 2016.

[24] Mallampati D., "An Efficient Spam Filtering using Supervised Machine Learning Techniques,"*International Journal of Scientific Research in Computer Science and Engineering*, vol. 6, no. 2, pp. 33-37, 2018.

[25] Manisha A. and Jain M., "Data Pre-Processing in Spam Detection," *International Journal of Science Technology and Engineering*, vol. 1, no. 11, p. 33-37, 2015.

[26] Manoharan S., Sugumaran P., and Kumar K., "Multichannel Based IoT Malware Detection System Using System Calls and Opcode Sequences," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 261-271, 2022.

[27] Mashaleh A., Binti Ibrahim N., Al-Betar M., Mustafa H., and Yaseen Q., "Detecting Spam Email with Machine Learning Optimized with Harris Hawks Optimizer (HHO) Algorithm,"*Procedia Computer Science*, vol. 201, pp. 659-664, 2022.

[28] Mccord M. and Chuah M., "Spam Detection on Twitter Using Traditional Classifiers," *in Proceedings International Conference on Autonomic and Trusted Computing*, Banff, pp. 175-186, 2011.

[29] Mishra R. and Thakur R., "Analysis of Random Forest and Naïve Bayes for Spam Mail using Feature Selection Categorization," *International Journal of Computer Applications*, vol. 80, no. 3, pp. 42-47, 2013.

[30] Quinlan J., Programs for Machine Learning, 1993.

[31] Rakotomalala R., "TANAGRA: a Free Software for Research and Academic Purposes," *in Proceedings of EGC*, Paris, pp. 697-702, 2005.

[32] Rusland N., Wahid N., Kasim S., and Hafit H., "Analysis of Naïve Bayes Algorithm for Email Spam Filtering Across Multiple Datasets," *in Proceedings of IOP Conference Series: Materials Science and Engineering*, Melaka, 2017.

[33] Sao P. and Prashanthi K., "E-mail Spam Classification Using Naïve Bayesian Classifier," *International Journal of Advanced Research in Computer Engineering and Technology*, vol. 4, no. 6, 2015.

[34] Shah N. and Kumar P., "A Comparative Analysis of Various Spam Classifications," *in Proceedings of Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*, Springer, pp. 265-271, 2018.

[35] Shams R. and Mercer R., "Classifying Spam Emails Using Text and Readability Features," *in Proceedings of the IEEE 13th International Conference on Data Mining*, Dallas, pp. 657-666, 2013.

[36] Sharaff A., Nagwani N., and Dhadse A., "Comparative Study of Classification Algorithms for Spam Email Detection," *in Proceedings of the Emerging Research in Computing, Information, Communication and Applications*, Springer, pp. 237-244, 2016.

[37] Shrivastava A. and Dubey R., "Classification of Spam Mail using Different Machine Learning Algorithms," *in Proceedings of the International Conference on Advanced Computation and Telecommunication*, Bhopal, pp. 1-10, 2018.

[38] Srinivasan S., Ravi V., Alazab M., Ketha S., Al-Zoubi A., and Kotti Padannayil S., "Spam Emails Detection Based on Distributed Word Embedding with Deep Learning," *in Proceedings of the Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, Springer, pp. 161-189, 2021.

[39] UCI Machine Learning Repository: Spambase Data Set. (n.d.), from http://archive.ics.uci.edu/ml/datasets/Spambase/, Last Visited, 2022.

[40] Vapnik V., *The Nature of Statistical Learning*, Theory, 1995.

[41] Whittaker S., Bellotti V., and Moody P., "Introduction to this Special Issue on Revisiting And Reinventing E-Mail," *Human-Computer Interaction*, vol. 20, no. 1-2, pp. 1-9, 2005.

[42] Yitagesu M. and Tijare P., "Email Classification Using Classification Method," *International Journal of Engineering Trends and Technology*, vol. 32, no. 3, pp. 142-145, 2016.

[43] Yüksel A., Cankaya S., and Üncü İ., "Design of a Machine Learning Based Predictive Analytics System for Spam Problem," *Acta Physica Polonica*, vol. 132, no. 3, 2017.

Hidayet Takci received his Ph.D. in Computer Science from Gebze Institute of Technology (Turkey) in 2005. He has been working as an Associate Professor at Sivas Cumhuriyet University Computer Engineering Department since 2011. He has 10 published research articles, 2 books, 4 book chapters and 10 scientific projects. His fields of study include Machine Learning, Natural Language Processing, Data Mining and Information Security.

Fatema Nusrat is a young researcher, currently serving as an instructor of Computer Science at the Asian University for Women, Chittagong, Bangladesh. She received the degree of Master of Science in Computer Engineering (2022) awarded by Konya Technical University, Turkey, and the degree of Bachelor of Science (Engineering) in Computer Science & Engineering (2016) from the International Islamic University Chittagong, Bangladesh. Her research interests include Data Science, Machine Learning, Deep Learning, and Neural networks.