Piotr ŚWITALSKI[1],
Mateusz KOPÓWKA[1]

1 Siedlce University of Natural Sciences and Humanities
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

Machine Learning Methods in E-mail Spam Classification

DOI: 10.34739/si.2019.23.04

Abstract. Increasing number of unwanted e-mails has influence on users' security in the Internet. Today spam e-mails can store potential malicious messages which e.g. can redirect user to fake sites. These messages recently appeared in social media. Filtering of this content is important due to minimize financial and branding costs. Traditional methods of spam filtering cannot be sufficient for present threats. We required new methods for constructing more dependable and robust antispam filters. Machine learning recently becomes very popular technique in classification methods. It has been successfully used in spam classification. In this paper we present some methods of machine learning for spam detecting. We would also like to introduce ways to solve the spam classification problem. We show that these methods can be useful in classification of malicious messages. We also compared developed methods and presented results in the experimental section

Keywords. spam detecting, machine learning, classification methods, spam filters, malicious messages

1. Introduction

Spam among Internet users is one of the best known terms. Every user who has e-mail account encounters this type of e-mail. They usually put negative emotions on the person. Spam is defined as a message that is not desired by the recipient. Mainly, it is sent to a large number of people, causing clogging of servers or blocking of mailboxes. Spambots, or senders of this

type of e-mail, constantly develop new spamming techniques to avoid the filtering that is built into e-mail services. The progress of technology significantly facilitates this procedure, and poor filtration exposes users to huge losses, not only material, but also personal, which significantly discourages people from using a given website and moving to another.

In the other hand we can meet with the term ham. It is a text message that is desired by the recipient but it is not considered as spam. There are two reasons why such a message is sent:

- directly - when you download free software, games or tools, or when registering for a new online service. In such cases, the user must check the box and accept the terms of service. There is also a field in which he agrees to receive advertising e-mails from the website and its partners. At that point, he actually legally agreed to receive this type of message,
- indirectly - it works in the same way as directly, but with one key difference. The information and offers field is already selected, and the user has the choice of deselecting it or not.

In [1] Ahmed and Abulaish show that spammers focus not only on access to e-mail, but also to social media. Most of the previous work on spam has focused on one specific e-mail only, while this type of spam is a relatively new area and the literature is little known. So they proposed a certain set of general statistical functions that were found on the basis of data from the most popular social networks, and then identified spam profiles on various types of social networks.

Users share information, links to articles or websites of interest to them. They also send each other files. Most of them rely on a direct group of trusted people and they are the ones users share most often. Spammers that exist in this society use such trusted networks to spread unwanted content by users. Many strategies have been developed to access such networks. One of them is URL shortening, which makes communication much easier. This function, due to its simplicity, is very harmful if it has links to advertisements or frauds, and other, such as phishing, for example, where the spammer tries to intercept user data [3].

Currently, the percentage of spam in e-mail traffic, as reported by Kaspersky Lab, in 2019 was 56.26% (which is 4.03 p.p. more than in 2018). The main source of spam comes from China 21.26%. 78.44% of spam e-mails were less than 2 kB in size. We need mention that over 15% of unique users encountered phishing. Phishing is one of the most important types of ”social engineering” attack. It can be disguised in many ways and used for different purposes. The trend of rapid growth of phishing attacks is a real problem. In Q2 2019 Kaspersky Lab has been recorded over 130 million phishing attacks.

These statistics show that malicious messages are a real problem for users. In one hand typical spam can be only irritating for users. It reduces bandwidth, which causes companies to lose a large amount of money due to lack of operation. In the other hand spam can be dangerous. Malicious messages are intended to accomplish the common purpose: to induce the user to do something, e.g. can be responding to the e-mail, clicking on an embedded link, or opening an attachment. Attackers usually focus on four primary human feature types: appeals to emotion (need of success, ambition, envy, narcissism), trickery (legitimate the way to get users to do something they may not normally do), subversive links (links that hide their true nature - usually they redirect user to fake websites), and subversive attachments (e.g. hide or change attachment file extensions, so can run itself as an application) [10].

Phishing messages generally contain an attractive information for a reader. Reader is persuaded to click a link or open an attachment. Attackers promise of an enormous payout an the end (the Nigerian scam) or exploit religious values to try the user to respond. These messages can invite the recipient to participate in some activity (the invitation seems very attractive to reader). For example it can offering special discounts for desirable and expensive products (smartphones, notebooks, etc.). In the other hand, malicious messages can warn recipients that their accounts have been compromised and they need immediately change their passwords and other login credentials. They usually are very convincing, so user click delivered in the message button or link. Some sort of e-mails masking the source of these message - its legitimate because they seem to come from well-known source. Another e-mails can use a confidence of people in so called ”a big lie”, e.g. safe 5G networks.

2. Methods of spam detecting

There are different spam detection strategies. The methods try to determine who among the senders of e-mails is a spammer and who is a legitimate sender. Moreover, spam exists not only in e-mail, but also in social networking sites. In recent years, such a network has been established in social media. They have become a common tool for communication and information exchange.

In order to filter spam, it is therefore necessary to distinguish this type of messages from those that the user actually wants to receive. So it is important to identify the typical characteristics of spam or malicious messages. The tactics used by spammers are constantly being improved, so it is good to know their new practices and effectively block their messages.

The characteristics of these e-mails are as follows:

- Headers - The headers usually show the route the e-mail took to get to the destination. It often also contains other information such as the sender, recipient, ID, date or subject. In the case of spam e-mails, the person sending the message tries to hide his identity, thus falsifying the headers. See an example of spam e-mail header presented in Fig. 1.

```
Received: from CORREO.ORIHUELA.ES ([127.0.0.1])
  by localhost (correo.aytoorihuela.com [127.0.0.1]) (amavisd-new, port 1032)
  with ESMTP id 0vLqcBF0v1fJ; Wed, 30 Sep 2020 10:49:29 +0200 (CEST)
Received: by 2002:a0c:c342:0:0:0:0:0 with SMTP id j2csp193229qvi;
        Wed, 30 Sep 2020 01:49:45 -0700 (PDT)
Reply-To: "Cristy Davis" <goedertsheryll1@outlook.com>
From: "Sheryll Goedert" <egmartinez@orihuela.es>
Subject: {Spam?} Darowizna
Date: Wed, 30 Sep 2020 10:48:29 +0200
Message-ID: <100659360.594706.1601455709894.JavaMail.zimbra@orihuela.es>
MIME-Version: 1.0
Content-Type: multipart/alternative;
                    boundary="----=_NextPart_000_0093_01D69718.C24AA750"
X-Mailer: Zimbra 8.8.6_GA_1906 (zclient/8.8.6_GA_1906)
Thread-Index: AQFp/bjXJCWgXoqF5z5NUyFGpzp4GA==
X-Google-Smtp-Source: ABdhPJwcvHQEKX0/9omOjmOPt0J27AfYdgq4bAo8fxO
0TXRtjAPd6qp8k4RSi3JEa/8wTCHkIzYE
X-Received: by 2002:a7b:c210:: with SMTP id x16mr1729635wmi.37.160455785816;
        Wed, 30 Sep 2020 01:49:45 -0700 (PDT)
Authentication-Results: mx.google.com;
       dkim=pass header.i=@orihuela.es
       header.s=DD20C93C-B4F2-11E9-9420-FB3EBDEFC590 header.b=vPQkI+MT;
       spf=pass (google.com: domain of egmartinez@orihuela.es
       designates 77.231.124.59 as permitted sender)
       smtp.mailfrom=egmartinez@orihuela.es;
       dmarc=pass (p=QUARANTINE sp=QUARANTINE dis=NONE)
       header.from=orihuela.es
X-Virus-Scanned: amavisd-new at aytoorihuela.com
X-Originating-IP: [192.168.100.68]
```

Figure 1. An example of spam e-mail header (own study)

- Message - spammers use a specific language in e-mails, thanks to which they can distinguish the message they send from others, and the typical words that are in it are as follows: free, limited offer, click here, no risk, lose weight, earn, for free [13]. The syntax in these sentences is also incorrect, there are infinitives and sometimes they are words that when combined together form a sentence that does not make complete sense. The text of the message may refer to recent events, e.g. COVID-19 disease. (see an example of spam message presented in Fig. 2).

Article [3] outlines three strategies to detect unsolicited spam e-mails. They are divided to three branches focusing on specific issues:

- Prevention - in this approach it is difficult to implement spam into the community tagging system by limiting access such as interfaces, such as Captcha (Completely Automated Public Turing test to tell Computers and Humans Apart), where it is a

verification that checks whether the system it has to do with a person or a program and other types of restrictions applied to the user's account.

```
--
cześć
Nazywam się pani Sheryll Goedert, wygrałem kumulację Powerball
o wartości 396,9 miliona dolarów w marcu 2020 roku i chcę przekazać
darowiznę w wysokości 3 000 000,00 €. Przekazuję ci tę darowiznę za
miłość, jaką mam do ludzkości i dla ciebie, aby pomóc ludziom dotkniętym
pandemią Covid-19 w twoim kraju.
skontaktuj się ze mną w sprawie żądania tej darowizny.
pozdrowienia
Sheryll Goedert
```

Figure 2. An example of spam message (own study)

- Detection - this approach shows us the use of machine learning in situations such as text classification or link analysis. It allows to treat the body, i.e. the body of the message, as a set of objects that have related attributes. In spam, messages are objects and attributes are headers. In internet spam, pages are objects, and attributes are various types of external links and page content.
- Demotion - this type reduces the visibility of content that may turn out to be spam. They are based on the rank system, giving the possibility to organize the system, tags or users, guided by the trust score.

Mostly prevention will be not enough, because user cannot be able to verify all messages correctly. Detection is more powerful. There are many classical methods of detecting typical spam.

Mail filters have been introduced to protect users from unsolicited messages received by spammers. Filters are usually built-in an e-mail servers. Before delivery messages are verified by the filter. Messages are checked against criteria that mean whether a given message is spam or not. Filters can be also implemented in clients. Users can use add-ons in client mail software like Mutt, Elm, Microsoft Outlook, Pine, Mozilla Thunderbird, Kmail. These plugins can be part of internet security software and usually this package consist firewall, anti-virus, child protection, privacy protection.

We can also use some kind of reputation. An example of relying on domain reputation, which is implemented in Google's Gmail system, was presented by Taylor in [14]. Reputation in this system is calculated based on the results of statistical filtration and the opinions of network users. Depending on the reputation domain is classified. In the case of a good reputation, the domain is whitelisted. However, if a domain has a negative reputation, then it ends up on the blacklist. If a message is sent from a domain that is not on any of the lists, it is

verified with statistical anti- spam filters to make the final decision. In addition, website users can send feedback if the classification was wrong. The topic of fake source addresses that negatively affect systems supported by senders is also discussed here.

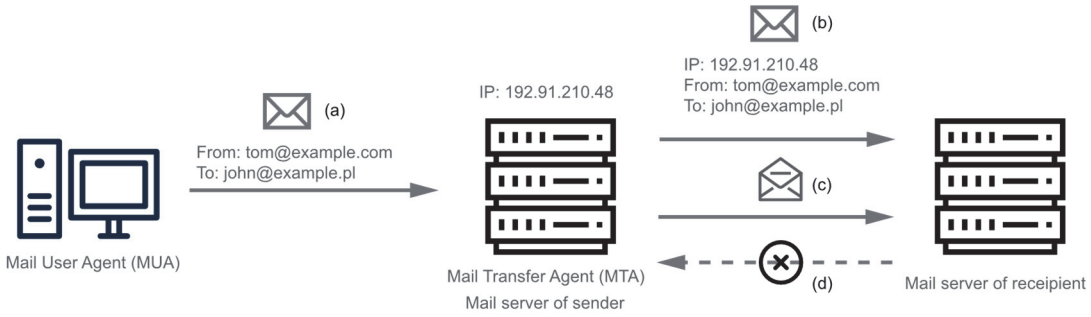

Figure 3. A system which applies Greylisting method (own study)

A heuristic method called Greylisting [6] was introduced by Harris to avoid spam by e-mail users. Let's consider a scheme of a system which applies Greylisting method - see Fig. 3. User using own Mail User Agent (MUA) sends a message to his MTA (Mail Transfer Agent) (see Fig. 3a). Receipment's MTA, which uses the gray list, when it receives an attempt to deliver a message, it will temporarily reject it, sending a notification of errors by SMTP (Simple Mail Transfer Protocol) (see Fig. 3b). Then the message is sending again (see Fig. 3c). When a message is received from senders who have been remembered before, it will be delivered. For spammers who focus on simplicity and speed. These people ignore any of the error messages. For this case the message won't be send (see Fig. 3d). They pass through to the next recipient sequentially, as opposed to trying to redeliver the message. By using this method, you can avoid unwanted messages that come from an unknown source.

These approaches are very often unreliable, and the filters that are applied or built into the system do not bring the expected results. The development of the classical approach is another step in the fight against unsolicited messages sent by people seeking to gain benefits. In order to improve spam filtering we can use more sophisticated methods.

Evolutionary algorithms build systems that adapt using a set of rules that use the principles of evolution. These are probabilistic search methods that have the ability to find solutions to a given search problem and optimize using a simulation of natural biological evolution. With each successive generation, the population evolves towards places in the search space that give better and better results. They use a simulation of the Darwinian evolution process.

The article [9] describes the spam classification and identification. The method can be limited to system filtering errors and spam detection by proposing a system that includes a set

of algorithms to extract functions. The genetic algorithm was used to find the optimal set of feature weights that will improve the classification accuracy. Another algorithm that was used was the immune system algorithm. It has a very similar structure to a genetic algorithm and therefore they complement each other. In both the genetic and immunological systems, search methods depend on a combination of deterministic and probabilistic rules. Due to their properties, genetic algorithms can be great for extracting traits from spam e-mails.

Training neural networks is one of the machine learning filtering techniques. It comes from the field of artificial intelligence and allows to learn and adapt to new types of e-mail that have similar properties to the ones they have previously learned. An artificial neural network is a very large number of algorithms and techniques that enable classification, regression or density estimation.

It has developed significantly over the years and many other types have emerged [7]. The perceptron and the multilayer perceptron are main types of neural networks that are used in this topic. The Neural Network Classifier is designed to detect spam and classify it using attributes based on the descriptive characteristics of the most unique patterns used by spammers. The article [11] describes the use of neural networks in the detection of malicious messages. This work showed that the neural network system is perfect for classifying such messages. However, in order to increase the precision of such a network, supervision is necessary.

Sharma, Prajapat and Aslam in [12] presented use of a multilayer perceptron neural network (MLP) and naive Bayesian model. In work [2] the multi-stage Neural Network was used to filter spam e-mails. As authors proved this technique outperformed Multi-Layer Perceptron (MLP) and perceptron classifiers.

Work written by Karthika and Visalakshi [8] propose an approach to use both Support Vector Machine (SVM) and Ant Colony Optimization (ACO) algorithms for spam classification. In this hybrid system SVM is used as the classifier. The feature selection of e-mails is implemented in the ACO algorithm.

3. Classification methods

Recently, classification methods developed in terms of machine learning becomes popular to detecting spam messages. These methods are automatic and adaptive, however mostly need learning stage. Most of the spam filtering methods are based on some text categorization algorithms. E-mails can be classified as either spam or ham using rules. So, the person who intend to spam filter must create set of rules. In machine learning methods, instead set of rules

we can use set of training samples which are pre-classified e-mail messages [4]. Classification process is provided by machine learning algorithms. To this algorithms belong Deep Learning, Naive Bayes, Support Vector Machines, Neural Networks, K-Nearest Neighbour, Rough sets, and Random Forests.

3.1. Linear Regression

Linear regression is one of the most popular algorithms in both statistics and machine learning. In machine learning, many algorithms are borrowed or reused. It is tested as a model to understand the relationships between input and output numeric variables. It is a linear model, e.g. having a model that contains a linear relationship between x input and a single output *y*, it can be calculated from a linear combination of *x* input. There are many different techniques to prepare and train a linear regression equation. Given the data that is very often called least squares, a model prepared in this way is often called least squares linear regression.

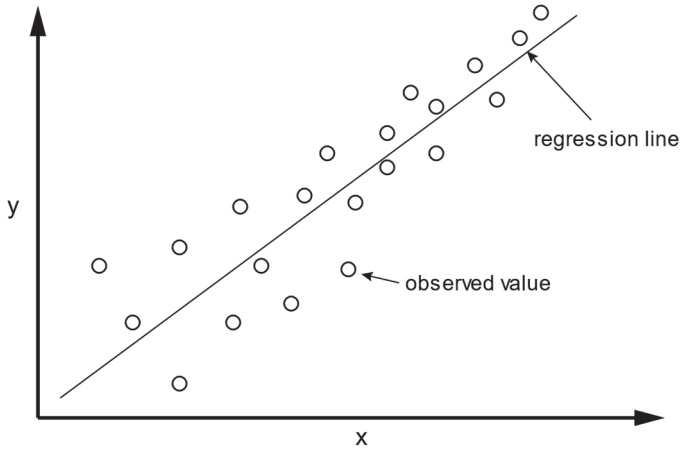

Figure 4. An example of linear regression solution (own study)

The representation of this model is a linear equation that combines a specific set of input *x* values, the solution of which is the predicted output for that set of input *y* values (see Fig. 4).

$$y = 0_0 + 0_1 x_1 + 0_2 x_2 + \cdots + 0_n x_n \tag{1}$$

where:
y - forecast value,
n - a number of features,
x_i - i-value - the feature,
0_0 - load point,
0_j - j-value - the weight of the feature.

3.2. Logistic Regression

Logistic regression is another technique that comes from statistics. The name comes from the function used in the core of the method. The logistic function is also called the sigmoid function. It can take any value in the real number range, and map it to values between 0 and 1, but it will never appear at these limits.

The model of logistic regression is estimated by:

$$P = \sigma(\Theta^T x), \tag{2}$$

where:
σ - logistic function,
Θ^T - transposed vector of weights,
x - vector of features.

3.3. Naive Bayes classifier

This classifier represents a fundamental probabilistic model. During machine learning, we are usually interested in the best choice of hypothesis a for specific data b. One of the simplest solutions to choosing the best hypothesis is given by the data we have. We can use this knowledge as knowledge to solve the problem. Bayes' theorem gives us the opportunity to calculate the probability of a hypothesis based on the knowledge we already have. This model is represented by following expression:

$$P(a|b) = \frac{P(b|a)P_a}{P_b} \tag{3}$$

where
P(a|b) - probability of event a, if event b occurs,
P(b|a) - probability of event b, if event a occurs,
P_a - probability that the hypothesis a is true,
P_b - probability of considering data.

Learning such a model from training data is very quick because you only need to calculate the probability of each class and the probability of each class with different inputs.

3.4. K-Nearest Neighbors

K-Nearest Neighbors is a simple algorithm that can classifies data according to a similarity measure. It is based on calculating of the nearest neighbors for each query point. There is a k parameter that refers to the number of nearest neighbors to include into majority set.

Let's consider an example in Fig. 5. We can see a new point marked with question mark which should classify to one of classes. First, we need find the *k* closest point to our new point and then classify points by majority vote of its *k* neighbors.

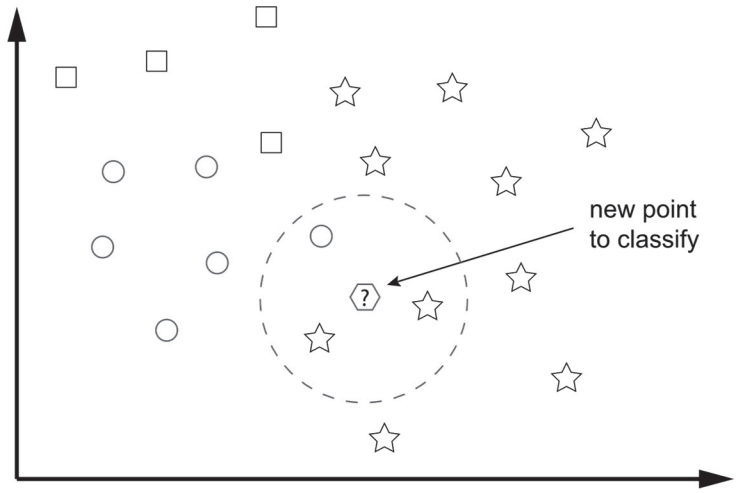

Figure 5. An example of K-Nearest Neighbors solution (own study)

Each point votes for their class and the class with the most votes is taken as the prediction. To find closest similar points, we need to calculate the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance or cosine distance. Then we get the nearest neighbors we wish to take the vote from according to *k* parameter. After voting a new point is assigned to class which was indicated by the majority.

3.5. Support Vector Machines

Support Vector Machines (SVM) is one of the most popular machine learning algorithms. Efficient and versatile used for linear, non-linear classification, regression and outlier detection. Numeric variables at the *x* input in the data form an n-dimensional space. If we have two input variables, a two-dimensional space will be created. SVM model constructs a hyperplane or set of hyperplanes (in multi-dimensional space).

The input data space is classified (divided by a hyperplane) into classes with some margin (see Fig. 6). A hyperplane, on the other hand, is a line that divides the input data space and allows for the best separation of points in the input variable space according to their class. The margin is calculated as perpendicular from the line to the nearest points. These points are called support vectors.

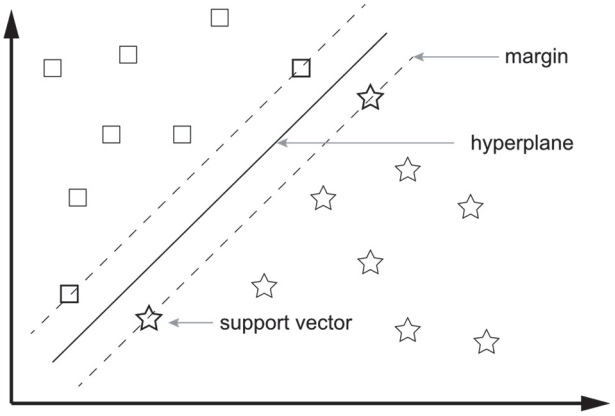

Figure 6. An example of Support Vector Machines solution (own study)

3.6. Neural networks

Artificial Neural Networks are groups of simple processing units (artificial neurons) which are interconnected, and communicate with one another by means of a sizable number of weighted connections [4]. Each processing unit can accept input from the neighbor. Next the output value is calculated and passed to other neighbors. The connections between units are weighted. By changing weights we can train the network. The mentioned unit can be considered as Threshold Logic Unit (TLU) or Perceptron. In TLU weighted signals are summed by simple arithmetic addition to provide node activation. The perceptron is an extension of TLU. The neurons in the perceptron are completely independent of each other, even the weights that make up each neuron are separated. So they do not have any connections with each other. The only thing they know about themselves is the input data, as each input can see all the data that enters the network.

4. Machine learning in filtering e-mail messages

To filtering e-mails using machine learning we need train the algorithms, so we need learning data. In such a way we can use datasets of real spam messages. Messages within datasets require transformation to be recognizable by the classification algorithm.

4.1. Text normalization

The raw text that is sent contains many useless characters or numbers that can significantly disturb the quality of the research being carried out. In this case, therefore, there is a need to process the text and find key words in the messages that will allow you to evaluate them and facilitate their classification. The process of converting data into a computer-readable form is often called pre-processing. One of the main forms of processing is to filter out unnecessary data. This process is provided by:

- removing whitespaces - each character or space is understood by the machine as some value that may somehow distort the result during classification,
- removing numbers - numbers are not a value that positively affects the examination of messages in terms of its classification. These are blank numbers that do not give any desired content in the e-mail,
- removing punctuation marks - punctuation marks, similarly to white spaces, in the opinion, also have values that may affect the final result of the classifier,
- minimal length of tokens - as each of the tokens is treated as a separate feature, their length matters. To do this, in order not to create unnecessary tokens that may affect fluctuations in results, its minimum length is two characters.

Text normalization can be defined as the transformation of all text into one form that it did not have before. This process makes it much easier to perform operations on it because the text is consistent.

4.1.1 Stopwords

Stopwords are usually the most common words in a language. They do not have any relevant information. These words are filtered by natural language data processing. There is no universal stop word list. The research process used the NLTK (Natural Language Toolkit) 2, which includes a set of libraries and programs for natural language processing.

4.1.2 Stemming

The next text normalization process is stemming. It is the result of the process of reducing changed forms or introduced words to their basic or source form. The stem need not be the same as the morphological root of a word. Texts in natural language have many different variants of the base of a word, e.g. connect, connection, connected, etc. In the search process, the problem of combining all these variants of a given word so that the query that has been asked determines only one variant.

4.1.3 Tokenization

It is the process of determining and classifying a string of input characters. When you have text, in this case an e-mail, you need to break it down into individual words called tokens. This is an essential process for text classification. To achieve a specific goal, it is necessary to understand the pattern of the text. Tokens are very useful for finding these patterns and are the basic step in stemming. An example of the sentence ”There are several reasons that asset tokenization is becoming so popular” is divided into tokens: ”There”, ”are”, ”several”, ”reasons”, ”that”, ”asset”, ”tokenization”, ”is”, ”becoming”, ”so”, ”popular”. Each word creates a separate token.

4.1.4 N-grams

N-grams are characters joined together with a length N. In the literature, the term may also include the concept of any character set in a string. A sentence is largely broken down into a set of n-grams that overlap. It is customary to analyze the entire text and count single occurrences, e.g. 1-gram - unigrams, double occurrences of 2-grams - bigrams or triples, 3-grams - trigrams (see Fig. 7).

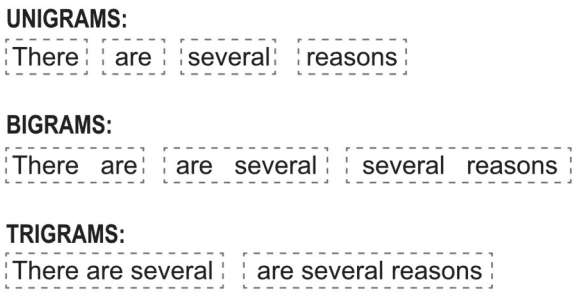

Figure 7. An example of unigrams, bigrams and trigrams.

The benefit of N-gram matching is due to its nature - a string called a string broken down into smaller pieces if it has any errors, it affects a limited number. In the case of counting N-grams that have common features for e.g. two strings, the value of their similarity is obtained and it is resistant to other errors in the text.

4.2. Performance evaluation measures

The performance measures are used to evaluate classification accuracy and performance of the spam detecting algorithms. Usually we compare a number of messages properly classified (as spam or not) or the percentage of messages properly classified. We should also take into account, how many messages are marked as false positives and false negatives. Message

marked as false positive is not spam, but is incorrectly being blocked as spam. It may lead to losing valuable information as a result of the spam detecting algorithm(s). The message is directly redirected to spam folder, so user may not notice it. In contrast, message marked as false negative is spam, but is incorrectly seen as a regular e-mail. This message is shown in received folder, so user can be irritated seen that message. However, this situation is less threatening for user.

Let's define some parameters:

- TP (True Positive) – true positive value,
- TN (True Negative) – true negative value,
- FP (False Positive) – false positive value,
- FN (False Negative) – false negative value.

To evaluate the classification algorithm we can use ROC-AUC. ROC curve is a curve which is a tool for assessing the correctness of the classifier operation. ROC (Receiver Operating Characteristic) is created by plotting the true positive TPR frequency (True Positive Rate) against the false positive FPR (False Positive Rate).

If we have many variables that affect the occurrence of an event, the goal is to use all the information they have, i.e. to build rules - classifiers that use many variables simultaneously. When these models are applied, a similarity to belonging to this class is obtained, i.e. scoring. In order to precisely define the appropriate criterion, measures of the quality of decision rules have been introduced. There are two main measures: specificity and sensitivity. They have the following patterns:

$$Sensitivity = \frac{TP}{TP+FN} \tag{4}$$

and

$$Specificity = \frac{TN}{TN+FP} \tag{5}$$

A popular approach is to calculate the area under the graph of this curve. This is known as AUC (Area Under Curve) (see Fig. 8). It is treated as the validity of a given model. The higher the value, the better the model.

5. Experimental results

The main problem with searching and finding spam messages is the binary character of classification task (whether it is spam or not). It is based on identifying the e-mail label whether

it is marked as spam or ham. The following classifiers were used to solve this problem: Naive Bayesian classifier, logistic regression, vectors of supporting machines, k-nearest neighbors and a neural network. The Scikit-Learn [31] library was used to conduct the research.

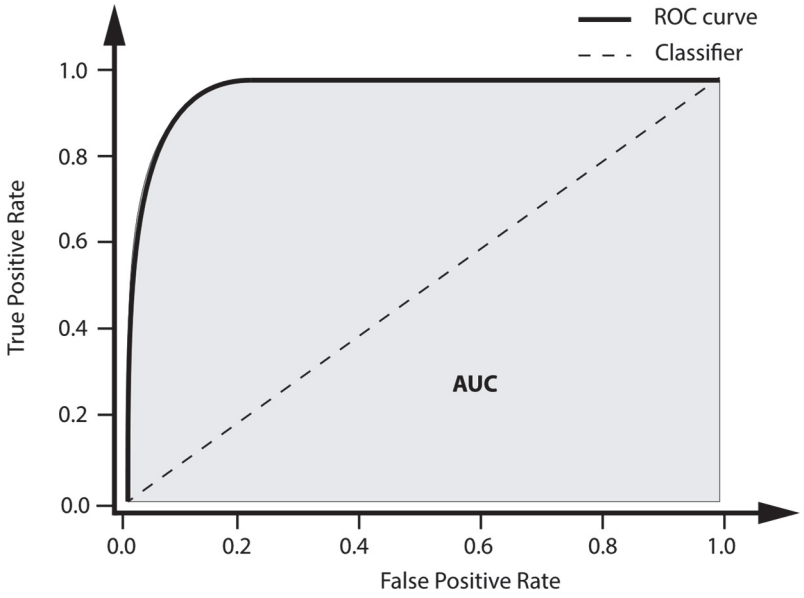

Figure 8. An example of ROC-AUC model (own study)

For experiments we have used Enron dataset. Due to the size and computing power, a sample of approximately 34,000 e-mails written in English has been selected. After removing duplicates and empty e-mails, the data collection counted 29,241 unique messages. In order to achieve the best results, the text needs to be processed. After normalization process, the words were assigned to values because the algorithm always expects the input data to be integers or floating point numbers. Therefore, a need arose to convert words to integers or floating point numbers. For this purpose, we determined of the set features and conversed of the set of unprocessed documents into the TF-IDF function matrix (Term frequency, inverse document frequency). The TFIDFVectorizer module from the Scikit-Learn library was used 3. TF-IDF is term frequency weighting - the inverse frequency in documents. This is one of the methods for calculating the weight of words converted into the number of occurrences. The frequency term summarizes how often it appears in a document, while inverse document frequency scales down the words that appear in different documents.

After loading the file with the e-mail message database, the program maps labels for messages, marking them as ”0” Ham message and ”1” as Spam message, then the pre-processing method is used to obtain the file structure that will be needed to convert the text into numbers integer or floating point. The previously mentioned Scikit-Learn library was used to create the program, which provides classifiers used in the research. In addition, Pandas libraries

were used to manage DataFrame objects, which is a data set, NumPy with the ability to calculate mathematical functions and NLTK to work with text. To evaluate the performance of the models, the Confusion Matrix was used, which enables the visualization of the correctness performance, cross-validation and the ROC-AUC curve.

The algorithm is presented in the following steps:

1. Load the Enron corpus dataset.
2. Label mapping, messages marked with "1" are spam messages and messages marked with "0" are Ham messages.
3. Checking if this set has empty fields and removing duplicate messages.
4. Text normalization in order to identify the most valuable words and convert them into tokens.
5. Split the data set into a test set and a training data set.
6. Convert the normalized text that was performed in step 4 to integers or floating point numbers with TFIDFVectorizer.
7. Create a classifier object with basic parameters and train it.
8. Calculation of the model accuracy.
9. Calculation of the average accuracy using the cross-validation.
10. Calculation of the accuracy of the ROC-AUC curve.
11. Repeating steps 7-10 for each classifier.

In the first experiment, the dataset was divided into two subsets (training set and testing set), then compared in terms of their effectiveness. The first experiment used 29,241 unique e-mails divided into 80:20, where 80% was a training set and the testing set was 20%. We also examined impact of use unigrams and bigrams to learning process. The purpose of this operation is to find out what combination can improve results. Firstly, we calculated ROC-AUC values. Examples of ROC-AUC plots for different classifiers are presented in Fig. 9.

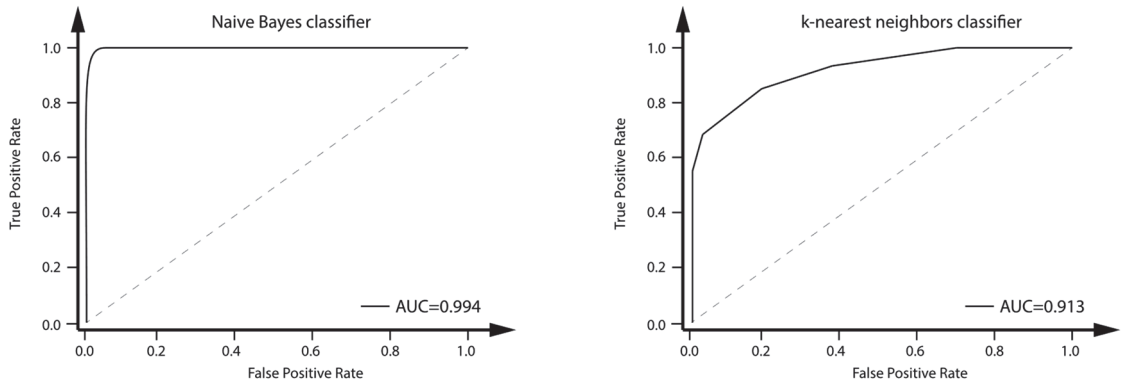

Figure 9. Comparison of ROC-AUC plots for different classifiers: Naïve Bayes (left) and k-nearest neighbor (right) obtained with use of bigrams (own study)

Then we calculated accuracy and cross-validation values. Results for data with use of unigrams in the 80:20 split are listed in Tab. 1.

Table 1. Best results for unigrams in the 80:20 split

Classifier	Accuracy	Cross-validation	ROC-AUC
Naive Bayes	98.37%	96%	99.83%
Logistic Regression	98.05%	98%	99.74%
Support Vector Machines	98.05%	99%	99.80%
K-Nearest Neighbour	96.78%	97%	98.88%
Perceptron	98.01%	98%	99.74%

For comparison we also examined identical subset with using of bigrams. The results for this experiment are listed in Tab. 2.

In this case results obtained by K-Nearest Neighbor were worse than values (about 30%) obtained by others classifiers. For the rest classifiers results were similar. One can see that results with use of unigrams are slightly better than using bigrams.

Table 2. Best results for bigrams in the 80:20 split

Classifier	Accuracy	Cross-validation	ROC-AUC
Naive Bayes	96.85%	97%	99.80%
Logistic Regression	97.09%	97%	99.40%
Support Vector Machines	97.65%	98%	99.65%
K-Nearest Neighbour	63.37%	64%	78.06%
Perceptron	96.44%	96%	99.59%

In the next experiment the dataset was split in proportion 70:30 (70% of training data and 30% testing data).

We checked how these proportions influence on the results. The results are listed in Tab. 3 and Tab. 4.

Table 3. Best results for unigrams in the 70:30 split

Classifier	Accuracy	Cross-validation	ROC-AUC
Naive Bayes	97.92%	98%	99.83%
Logistic Regression	97.61%	98%	99.69%
Support Vector Machines	98.33%	99%	99.79%
K-Nearest Neighbour	97.45%	98%	99.51%
Perceptron	97.87%	98%	99.76%

Table 4. Best results for bigrams in the 70:30 split

Classifier	Accuracy	Cross-validation	ROC-AUC
Naive Bayes	96.80%	97%	99.77%
Logistic Regression	97.15%	97%	99.38%
Support Vector Machines	97.70%	98%	99.62%
K-Nearest Neighbour	63.58%	64%	77.30%
Perceptron	95.93%	97%	99.56%

As we can see, changing of the training data size has minimal importance for the accuracy of the algorithms. In this case, two subsets that differed by 10% did not show any significant difference in the case of classifiers. However, properly scaling of the data can improve the results. In this case, normalization has a major impact on the obtained results.

6. Conclusion

We have presented several classification methods and methods of text normalization. We also compared efficiency of selected classification methods.

The research was carried out using two instances. The first was the 80:20 split of the data and the second was in the 70:30 split. This was to check how big the difference will be between the two studies in relation to each of the classifiers. Finally, the best results for unigrams and bigrams for each of the classification algorithms were summarized. It can be observed that the expected results of the Naive Bayesian Classifier and Artificial Neural Networks did not show that they give the best possible results in the classification. Perhaps it was influenced by the data that was tested and its size. Choosing the right features and good text processing is also important. First of all, neural networks did not show their superiority over other classifiers.

The use of machine learning classifiers in the problem of spam message detection gave a result of about 95-98%, which is a satisfactory result. The best result for the division of 70:30 for both unigrams and bigrams was given to the Support Vector Machines classifier. On the other hand when the 80:20 split was used, the Naive Bayesian Classifier (for unigrams) performed the best results. Also when we used bigrams, the previously mentioned Support Vector Machines classifier performed relatively better results.

What is important, each of the classifiers in the ROC curve achieved the area under the curve (AUC) close to 99%, which is a very good result and proved that the quality of the model is high.

In the presented research, machine learning algorithms offer many possibilities and can clearly have an effect in the fight against unwanted messages in e-mail. Development and research in this direction can definitely improve the quality of protection systems.

References

[1] Ahmed F., Abulaish M.: A Generic Statistical Approach for Spam Detection in Online Social Networks, Computer Communications, Volume 36, No 10–11, pp. 1120-1129, 2013, doi.org/10.1016/j.comcom.2013.04.004.

[2] Alkaht I.J., Al-Khatib B.: Filtering SPAM Using Several Stages Neural Networks, International Review on Computers and Software, Volume 11, No 2, pp. 123-132, 2016, doi.org/10.15866/irecos.v11i2.8269.

[3] Balogun A. K., Jaafar A., Murad M. A. A., Ezema E.: Spam Detection Approaches and Strategies: A Phenomenon, Foundation of Computer Science, Volume 12, No 9, pp. 13-18, 2017, doi.org/10.5120/ijais2017451728.

[4] Dada E. G., Bassi J. S., Chiroma H., Abdulhamid S. M., Adetunmbi A. O., Ajibuwa O. E.: Machine learning for email spam filtering: review, approaches and open research problems, Heliyon Volume 6, No 6, 2019, doi.org/10.1016/j.heliyon.2019.e01802.

[5] Gangavarapu, T., Jaidhar, C.D. and Chanduka, B.: Applicability of machine learning in spam and phishing email filtering: review and approaches. Artifical Intelligence Review 53, pp. 5019–5081 (2020), doi.org/10.1007/s10462-020-09814-9.

[6] Harris E.: The Next Step in the Spam Control War: Greylisting, Puremagic Software, 2003 (http://projects.puremagic.com/greylisting, access date: 04.10.2020).

[7] Jalab H. A., Subramaniam T., Taqa A. Y.: Overview of textual anti-spam filtering techniques, International Journal of Physical Sciences, Volume 5, No 12, pp. 1869-1882, 2010.

[8] Karthika R., Visalakshi P.: A hybrid ACO based feature selection method for email spam classification, WSEAS Transaction on Computers, Volume 14, 2015.

[9] Razi Z., Asghari S. A.: Providing an Improved Feature Extraction Method for Spam Detection Baased on Genetic Algorithm in an Immune System, Journal of Knowledge-Based Engineering and Innovation, Volume 3, No 8, 2017.

[10] Julie J.C.H.R. and Kamachi C.: Detecting and Combating Malicious Email Syngress, Chapter 2. Types of Malicious Messages., 2015, doi.org/10.1016/B978-0-12-800110-3.00002-2.

[11] Ndumiyana D., Magomelo M., Sakala L. Ch., Spam Detection using a Neutral Network Classifier, Online Journal of Physical and Environmental Science Research, Volume 2, pp. 28-37, 2013.

[12] Sharma A.K., Prajapat S.K., Aslam M., A comparative study between naıve Bayes and neural network (MLP) classifier for spam email detection, International Journal of Computer Applications, 2014.

[13] Spykerman M., Typical spam characteristics How to effectively block spam and junk mail, Red Earth Software, 2003.

[14] Taylor B., Sender Reputation in a Larger Webmail Service, CEAS 2006 - Third Conference on Email and Anti-Spam, 2006.