

Gaussian Radial Basis Function Neural Network with Correlation Based Feature Selection Applied to Medical Text Categorization

Akın Özçift*

Department of Software Engineering, Hasan Ferdi Turgutlu Technology Faculty, Manisa Celal Bayar University,
45400, Turgutlu, Manisa, Turkey
*akin.ozcift@cbu.edu.tr

Received: 03 October 2018

Accepted: 07 February 2019

DOI: 10.18466/cbayarfbe.466908

Abstract

Text categorization is an important field for information processing systems. Particularly, medical text processing is a popular research area that makes use of classification algorithms and dimension reduction strategies from machine learning field. In this study, we propose a three stage algorithm to automatically categorize medical text from OHSUMED corpus. In the proposed algorithm, we use Correlation Based Feature Filtering on top of Radial Basis Function Neural Network. The algorithm for 12 sample datasets produces 0.890 in terms macro average F-measure. In this context, both Correlation based Feature Filtering as a feature elimination strategy and Radial Basis Function Neural Network as text categorization algorithm are promising methods.

Keywords: Machine learning, text categorization, neural networks, feature selection

1. Introduction

The task of assigning free-text documents into one or more predefined categories is known as Text Categorization (TC). This task assumes availability of pre-categorized document corpus as training data to make prediction for the categorization of future instances. The process may include binary or multi-label categories. In binary labeling there are two distinct cases: (i) True, the document belongs to a specific category or (ii) False, the document is unrelated to that category [1]. Many present information management tasks, i.e. document dissemination, real-time file classification, Web page classification and topic identification, make use of TC methodologies [1]. In particular, medical information processing technologies and medical research generate rapidly increasing knowledge to be processed in some way. As an example, MEDLINE incorporates over 4600 biomedicine and health related journals from 1966 to the present. MEDLINE collection covers journal articles, and biomedical case studies about nursing, dentistry, veterinary medicine, and biological-physical sciences [2, 3]. Effective organization of such massive information is one of the important problems of TC to propose solution.

Text Categorization methodologies typically include four main stages: (i) collection of texts, (ii) pre-processing of documents, (iii) dimension reduction and finally, (iv) categorization of documents into pre-defined groups. The general lay-out of the workflow is provided in Figure 1.

- 1-> Collect documents
- 2-> Pre-process documents
 - i-Remove stop words and stop tokens
 - ii-Normalization
 - iii-Make stemming
 - iv-Determine weights of the terms from (iii)
 - v-Make pruning for less frequent words
- 3-> Reduce dimension with feature elimination
- 4-> Categorize documents into related groups

Figure 1. Text Categorization general workflow stages.

Automatic categorization of texts is implemented with the use of machine learning algorithm. On the other hand, TC process generates high dimensional data, and it is widely discussed in machine learning literature that classification (or categorization) of such data has two main problems; decrease in classification accuracy and high computational cost with immense memory usage [4]. Though, accuracy increase is not guaranteed with dimension reduction, at least computational complexity makes dimension reduction a preliminary step in TC. Dimension reduction or feature selection techniques score each potential feature with respect to a particular metric, and then generate the best sub-set of features. The two successful feature elimination methods in TC are Information Gain and Chi-Squared [5].

Many classification algorithms of machine learning are adopted for TC purposes. A few widely used TC algorithms are Decision Trees, Naïve Bayes, Bayes Nets, Support Vector Machines, Ripper and k-Nearest Neighbor [6].

The focus in this paper is to design a text categorization algorithm, i.e., Radial Basis Function Neural Network (RBF) with Correlation Based Feature Selection (CFS), applied to a part of MEDLINE documents.

2. Materials and Methods

In this section, we provide step by step explanation of the proposed algorithm depending on the context of the TC literature.

2.1. Definition of Text Categorization Problem

TC problem is defined as the task of assigning a Boolean value (*true*: document belongs to the category, *false*: document does not belong to the category) to the pair $\langle d_j, c_i \rangle \in D \times C$. In this notation, D represents a domain of documents $D = \{d_1, \dots, d_n\}$ to be categorized and $C = \{c_1, \dots, c_{|C|}\}$ corresponds to the predefined text categories for $|C| = m$. Formally, “the task is to approximate the unknown *target function* $\Omega: D \times C \rightarrow \{T, F\}$ (that describes how documents ought to be classified) by means of a function $\Omega: D \times C \rightarrow \{T, F\}$ called the *classifier*, i.e., *rule*, or *hypothesis*, or *model*” such that Ω^{\rightarrow} and Ω to be coincided as much as possible” [7]. There are two assumptions in this scheme; the categories have no additional meaning other than symbolic labeling and no extra knowledge such as document metadata (type, author and source) is available. Depending on this formalization, a classifier is first trained with the use of labeled (categorized) texts and the effectiveness of the model is next evaluated with previously uncategorized (unseen) instances [8].

2.2. Feature Representation and Pre-Processing

Machine learning algorithms cannot directly categorize documents and therefore the first step in text categorization is to transform documents into a special format called vector space model [9]. In this model, the structure of the document and the order of words in the document are disregarded. The feature vectors in this matrix representation are the words observed in the documents. The final word-list (features in the vector space model) $W = \{w_1, \dots, w_n\}$ in the training set is obtained through a set of pre-processing steps, i.e., tokenization, normalization, stop-word removal, stemming and removal of less-frequent words [7,10]. In this section, we briefly provide the general workflow of these steps with the use of a sample short-text from [11]: “The success of RF algorithm makes it eligible to be used as kernel of a wrapper feature subset evaluator. We used best first search RF wrapper algorithm to select optimal features of four medical datasets: colon cancer, leukemia cancer, breast cancer and lung cancer. We compared accuracies of 15 widely used classifiers trained with all features versus to extracted features of each dataset”.

i) *Tokenization*: Tokenization is to break-down a string (maybe a whole document) into meaningful constituents, i.e., words, phrases, symbols etc., called *tokens*. Since all contiguous alphabetic characters and numbers are part of one token, there are some heuristics algorithms to convert a string chain into tokens. In English language, words (or tokens) are in general delimited by whitespace or punctuation. From this point of view, tokenization of English texts is rather straightforward and some of the algorithms used for this transformation are alphabetic, n-gram and word tokenizers [12]. The sample text is already tokenized and we ignore to present it once more.

ii) *Normalization*: At this stage, the characters in training text are converted into either uppercase or lower case. Furthermore, punctuations and numbers are removed from text [13]. The sample text is provided in Figure 2.

the success of rf algorithm makes it eligible to be used as kernel of a wrapper feature subset evaluator we used best first search rf wrapper algorithm to select optimal features of four medical datasets colon cancer leukemia cancer breast cancer and lung cancer we compared accuracies of widely used classifiers trained with all features versus to extracted features of each dataset

Figure 2. Normalization of sample text.

iii) *Stop-world removal*: Stop-words, in the context of text categorization, correspond to non-informative words present in the documents. Removal of stop-words increases accuracy of prediction while reducing computational cost. Particularly, each unremoved stop word will result an extra irrelevant feature vector that will definitely increase computational load. In English language, non-informative words consist of about 400 terms that include articles, conjunctions, prepositions and some high frequency words. Some example stop-words are ‘the’, ‘and’, ‘as’, ‘it’, ‘first’, and ‘be’ [14].

Stop-word list are user defined and might be refined with respect to the application need. We apply stop-word removal process and we present the generated output in Figure 3.

One should note that, the sample text in normalization step includes 62 words and after stop-word process the text has only 41 words. This in practice means removal of 21 irrelevant feature vectors that do not describe the text. success rf algorithm makes eligible used kernel wrapper feature subset evaluator used search rf wrapper algorithm select optimal features four medical datasets colon cancer leukemia cancer breast cancer lung cancer compared accuracies widely used classifiers trained features versus extracted features dataset

Figure 3. Stop-word removal applied to sample text.

iv) *Stemming*: Stemming may also be defined as the *word normalization* that denotes the process of obtaining basic forms of the words, i.e. stems [15]. For instance, the common stem of words ‘used’, ‘use’, ‘useful’ is obtained to be as ‘use’ with the use of a stemmer algorithm. Stemming process makes use of stem dictionaries as basis and some stemming algorithms from literature are Lovins, Iterated Lovins, Krovetz and Snowball stemmers [15]. With the use of Lovins stemmer algorithm, the sample text is transformed into Figure 4.

succes rf algorithm mak elig us kernel wrapper featur
 subses evalu us search rf wrapper algorithm select
 optim featur four med datases colon cancer leukem
 cancer breast cancer lung cancer compar accur wid us
 classif train featur versus extract featur datases

Figure 4. Lovins stemmer algorithm applied to sample dataset.

Stemmer algorithms may generate some interesting results. For instance, as Figure 4 is examined, it is seen that word ‘use’ is transformed into ‘us’ as a common stem of the text. Though, semantically this output is unimportant for a machine learning algorithm, the text analyst should decide benefit of this transformation. The effect of stemmer algorithms on the text categorization performance is studied in [15] with detail.

v) *Pruning less-frequent words*: As it is briefly described in Section 2.2, Vector space representation comprises word vectors (features) through pre-processing of documents. Once the set of terms are obtained in a document, the goal is to transform these nominal values into numeric frequencies. In this scheme, “term weighting is applied to set a level of contribution of a term to a document” [16]. Term weighting can simply be realized with (i) word occurrence frequency, and (ii) binary term frequency approaches. For document set $D = \{d_1, \dots, d_n\}$, a document d_j is represented with $\vec{d}_j = \{w_{1j}, w_{2j}, \dots, w_{mj}\}$. In this notation, w_{ij} denotes the weight assigned to term t_i in the document. For unique set of terms, word occurrence counts the number of the specific term in the related document where as binary weighting uses 1 (for present term) and 0 (for non-present term). After the word weighting process, for instance, the terms with occurrences lower than 2 may be neglected. We applied word occurrence frequency strategy with 2 as pruning threshold to our sample text and we obtained the vector space representation in Table 1.

Table 1. Pruning less-frequent words process applied to sample text.

The Term	Frequency	The Term	Frequency
algorithm	2	rf	2
cancer	4	us	3
datases	2	wrapper	2
featur	4	category	yes

In Table 1, a ‘yes’ under *category* label means that the processed text belong to a document category organized for cancer diseases.

2.3. Data Dimension Reduction

In previous section, we made use of a short sample from cancer diagnosis domain while presenting the pre-processing steps. However, most of the time real applications produce thousands of word vectors that may cause problems for machine learners. In order to decrease dimension of the processed documents, feature ranking techniques (filtering algorithms) are used. Filtering algorithms evaluate the efficiency of features based on some relevance measure and this relevancy is designed to measure the dependency between class label and feature. In filtering approach, while the feature vectors being most relevant in the class prediction are retained, the remaining features are omitted. Some widely used filtering algorithms based on relevancy measure concept are Information Gain (IG), Gain Ratio (GR) and Chi-square [7, 17]. We provide brief information about these algorithms as follows:

“Information gain measures the amount of uncertainty associated with the class y that can be gained, given the knowledge of the value of feature x_i . The amount of uncertainty associated with the class y is measured by its entropy, defined as $Entropy(y) = -\sum_{l=1}^n \Pr(y = c_l) \log \Pr(y = c_l)$ [17]. Furthermore, the mathematical definition of information gain of a feature x_i is defined with $IG(x_i) = Entropy(y) - Entropy(y|x_i)$. In this notation, $Entropy(y|x_i)$ is the conditional entropy of y given x_i [17].

Gain ratio also makes use of *entropy* concept in its feature relevancy measure. Using previous entropy definition, GR may be defined as $GR(x_i) = \frac{IG(x_i)}{Entropy(x_i)}$.

Relevancy measure of Chi-square filtering algorithm is the lack of independence measure between class y and feature x_i . After this brief introduction to feature filtering algorithms, we provide information about correlation based filtering algorithm, i.e., Correlation Based Feature Selection (CFS), which we preferred in our text categorization system.

Correlation based filtering uses a search strategy while it evaluates merit of feature subsets. With more clear terms, CFS measures the usefulness of individual features in predicting the class label along with the level of inter-correlation among them. This logic originates from the hypothesis that relevant feature subsets contain features highly correlated with the class, yet uncorrelated with each other [18]. The merit of a feature subset S comprising k features may be given as

$Merits_k = \frac{k\bar{r}_{cf}}{\sqrt{k+k(k-1)\bar{r}_{ff}}}$. In this notation, \bar{r}_{cf} is the average value of all feature-classification correlations, and \bar{r}_{ff} is the average value of all feature-feature

correlations [18]. In our implementation, we used CFS algorithm with best first search strategy while identifying relevant subset of features.

2.4. OHSUMED Corpus

In this research, to evaluate the proposed two-stage algorithm, we use subset of OHSUMED corpus, which is widely used in medical text categorization studies as a benchmark. OHSUMED is a subset of the MEDLINE database, which is a bibliographic database of important, peer-reviewed medical literature maintained by the National Library of Medicine. This widely used test collection is from the OHSUMED corpus. The original OHSUMED corpus has 90 medical categories and in this research we make use 9 categories as benchmark for the proposed algorithm. The 9-category dataset is compiled by Yubin Kim [19]. The statistical properties of the dataset are provided in Table 2. Furthermore, for the sake of convenience we provide number of the features retained after feature filtering process.

Furthermore, the pre-processing steps of the datasets are as follows:

- i) Normalization: The texts are tokenized and each token is converted to lowercase.
- ii) Stop-word removal: 450 stop-words are removed from documents to remove uninformative word features from output.
- iii) Stemming: The token list from step (ii) is removed with the use of Krovetz stemmer to produce final word vectors. Furthermore, the feature values are defined with term frequency and no base threshold is selected for term frequency. The term frequency obtained in step (iii) is maintained as it is.

Table 2. Statistical properties of datasets and their filtered correspondings.

No	Name	Instances	Feature Size	Retain Features
1	Hyperplasia	241	4922	31
2	Mitosis	124	3545	28
3	Necrosis	481	7108	36
4	Pediatrics	127	3465	22
5	Plasmids	591	8042	71
6	Systole	135	3490	29
7	Pregnancy-	301	5686	37
8	Pregnancy-	1603	12697	56
9	Rats-I	301	5688	38
10	Rats-II	1603	12933	76
11	Prognosis-I	301	5389	40
12	Prognosis-	1603	12712	57

2.5. Normalized Gaussian Radial Basis Function Neural Networks

In this section, we provide RBF as our text categorization algorithm in brief terms. To the best of our knowledge, RBF is the first time used in OHSUMED domain.

A basic RBF architecture comprises three different neuron layers: (i) input layer consists of source nodes, (ii) hidden layer neurons with Gaussian radial basis functions to compute outputs and (iii) output layer to build a linear weighted sum of hidden neuron outputs [20]. The output of neural network $\phi: R^n \rightarrow R$ is defined as $\phi(x) = \sum_{i=1}^N a_i \rho(\|x - c_i\|)$. In this notation, N denotes the number of neurons in the hidden layer, c_i is the center vector for i th neuron, and a_i are the weights of the linear output neuron. The weights c_i and a_i are obtained under the condition that they optimize the fit between ϕ and the data. However, the above RBF definition is un-normalized RBF. In case of normalization, the definition of the network may be rewritten as $\phi(x) = \sum_{i=1}^N a_i \theta$ and in this definition θ is given by $\theta = \frac{\rho(\|x - c_i\|)}{\sum_{i=1}^N a_i \rho(\|x - c_i\|)}$.

In addition to RBF, we make use of Naïve Bayes (NB), Support Vector Machines (SVM), k-Nearest Neighbor (KNN), Ripper (RIP) and C45 as benchmarking algorithms to evaluate performance of RBF algorithm.

2.6. Statistical Evaluation Metrics

In text categorization literature, the evaluation metrics used frequently are Precision (Pr), Recall (Rc) and F_1 measure (or simply F-measure). F_1 is defined as the harmonic mean of Pr and Rc, and in this concept Pr and Rc are defined as $Pr_i = \frac{TP_i}{TP_i + FN_i}$ and $Rc_i = \frac{TP_i}{TP_i + FP_i}$. In this definition, TP_i corresponds to the number of texts assigned correctly to category i and FP_i (false positives) is the number of texts incorrectly assigned to the category i by classifier. Furthermore, FN_i (false negatives) defines the number of documents categorized under class i but actually belong to class i . Furthermore, F_1 measures performance of a text categorization algorithm with $F_1 = \frac{2PrRc}{Pr+Rc}$ by combining Pr and Rc in a single quantity. F_1 is accepted to be better as it approaches 1. One another additional metric that used in text categorization literature is macro-averaged F-measure. Macro-averaged F-measure (MF1) is the average of the F_1 values computed over each category.

2.7. The Algorithm

In this section, we provide workflow of the proposed algorithm in Figure 5 below.

3. Results and Discussion

We provide experimental results of the proposed algorithm in Table 3. One should note that the corresponding results are based on filtered feature sizes given in Table 2. On the other hand, we make use of Information Gain (IG) filter-ranker as a comparison base

for CFS algorithm. Furthermore, we make experiments with additional algorithms (NB, SVM, kNN, Ripper, C45) from text categorization literature to compare them with RBF.

```

1:
Text_Categorization
2:
3:
Data_Labeln ← N
4:
for n=1 do Label_n
5:
Lower_Case(Data_Labeln)
6:
RemStpWords(Data_Labeln)
7:
Krovetz_Stemming(Data_Labeln)
8:
End for
9:
Correlation_Based_Feature_Filter(Data_Labeln)
10:
Categorize_Data(Data_Labeln)
11:
F-measure = Evaluate_outcome(Data_Labeln)
12:
End procedure
  
```

Figure 5. Text categorization algorithm.

For the three best classifiers, namely RBF, NB and SVM, we provide Figure 6 to compare their performances in terms of F-measure for 12 datasets. It should be noted that we make use of corresponding numbers from Table 3 instead of dataset names.

4. Conclusion

There are many application fields of text categorization algorithms and dimension reduction methods. In particular, medical text categorization makes use of classification and dimension reduction strategies. Newly proposed algorithms are fundamental for better categorization of generated medical information. In this concept, RBF and CFS are first time used in medical text categorization field.

Experimental results demonstrate that RBF is slightly better than widely used SVM and NB algorithms in terms of F-measure. In this aspect, it may be an alternative categorizer algorithm for text analysis domain.

Table 3. Experimental results of the proposed algorithm.

No	Category	Feature Elm.	NB	RBF	SVM	KNN	RIP	C45	Macro Avg.
1	Pediatory	CFS	0.874	0.927	0.892	0.943	0.91	0.91	0.909
		IG	0.909	0.928	0.819	0.869	0.885	0.877	0.881
2	Mitosis	CFS	0.918	0.951	0.863	0.882	0.857	0.858	0.888
		IG	0.866	0.911	0.844	0.836	0.819	0.85	0.854
3	Systole	CFS	0.97	0.985	0.963	0.941	0.941	0.963	0.96
		IG	0.948	0.956	0.956	0.918	0.97	0.97	0.953
4	Hyperplasia	CFS	0.884	0.908	0.861	0.892	0.836	0.849	0.871
		IG	0.854	0.879	0.808	0.813	0.831	0.844	0.838
5	Prognosis-I	CFS	0.924	0.914	0.91	0.84	0.88	0.864	0.888
		IG	0.88	0.877	0.89	0.81	0.877	0.86	0.865
6	Pregnancy-I	CFS	0.838	0.862	0.841	0.814	0.804	0.79	0.824
		IG	0.814	0.845	0.823	0.784	0.804	0.788	0.809
7	Rats-I	CFS	0.937	0.937	0.944	0.887	0.927	0.94	0.928
		IG	0.94	0.91	0.94	0.89	0.933	0.93	0.923
8	Necrosis	CFS	0.847	0.748	0.806	0.836	0.809	0.81	0.809
		IG	0.759	0.821	0.821	0.736	0.803	0.785	0.787
9	Plasmids	CFS	0.927	0.936	0.942	0.869	0.91	0.907	0.915
		IG	0.924	0.937	0.92	0.842	0.905	0.904	0.905
10	Pregnancy-II	CFS	0.833	0.81	0.832	0.832	0.844	0.829	0.83
		IG	0.836	0.84	0.845	0.771	0.839	0.822	0.825
11	Prognosis-II	CFS	0.89	0.82	0.898	0.829	0.905	0.906	0.874
		IG	0.88	0.881	0.904	0.835	0.898	0.895	0.882
12	Rats-II	CFS	0.95	0.932	0.94	0.858	0.934	0.928	0.923

		IG	0.934	0.934	0.943	0.871	0.938	0.938	0.926
	Macro Avg.		0.889	0.890	0.883	0.849	0.877	0.875	

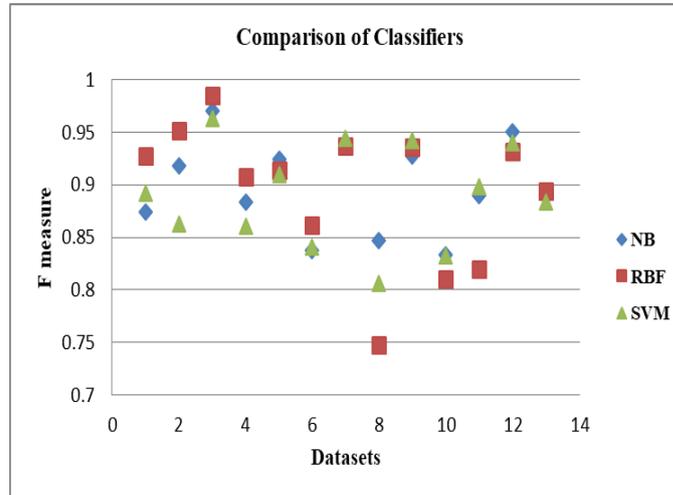


Figure 6. Comparison of the best three classifiers.

Feature dimension reduction, in general, is one of the preliminary steps in text categorization. In this context, feature elimination algorithms are useful to improve text categorization performance. As experimental results demonstrate, CFS is successful compared to IG in terms of macro average measure. Therefore, it may be an appropriate feature elimination algorithm for text categorization purposes.

Ethics

There are no ethical issues after the publication of this manuscript.

References

- Pons, A, Gil, P, Garcia, R, Berlanga, L. 2007. Using Typical Testors for Feature Selection in Text Categorization. *Lecture Notes in Computer Science, Springer*; 643-652.
- Qirui, Z, Jinghua, T, Huaying, Z, Weiye, T, Kejing, H. Machine Learning Methods for Medical Text Categorization. *Circuits, Communications and Systems, Pacific-Asia Conference, 2009*, pp 494-497.
- Yang, Y, Joachims, T. 2008. Text Categorization. *Scholarpedia Text Categorization*; 4242-4245.
- Janecek, A, Gansterer, W. On the Relationship Between Feature Selection and Classification Accuracy. *JMLR: Workshop and Conference Proceedings, 2009*, pp 90-105.
- Forman, G. 2007. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Resources*; 1289-1305.
- Deng, Z, Tang, S, W, Zhang, M. 2005. An Efficient Text Categorization Algorithm Based on Category Memberships. *Fuzzy Systems and Knowledge Discovery*; 480-485.
- Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*; 34: 1-47.
- Dumais, S. 1998. Using SVMs for Text Categorization. *IEEE Intelligent Systems*; 13: 21-23.
- Liao, Y, Vemuri, V, R. Using Text Categorization Techniques for Intrusion Detection. *Proceedings of the 11th USENIX Security Symposium, 2002*, pp 51-59.
- Li, Y, H, Jain, A, K. 1998. Classification of Text Documents. *The Computer Journal*; 41: 537-546.
- Ozçift, A. 2011. Enhanced Cancer Recognition System Based on Random Forests Feature Elimination Algorithm. *Journal of Medical Systems*; 1-9.
- McNamee, P, Mayfield, J. 2004. Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval*; 7: 73-97.
- Schapire, R, Singer, Y. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*; 135-168.
- Mendez, J, Iglesias, E, Riverola, F, Diaz, F, Corchado, J. 2006. Tokenizing, Stemming and Stopword Removal on Anti-spam Filtering Domain. *Current Topics in Artificial Intelligence*; 449-458.
- Text-Mining Research Group, University of West Bohemia, Influence of Word Normalization on Text Classification. <http://textmining.zcu.cz/publications/inscit20060710.pdf> (accessed at 10.01.2018).
- Lernattee, V, Theeramunkong, T. 2007. Effects of Term Distributions on Binary Classification. *IEICE Transactions on Information and Systems*; 1592-1600.
- Chou, C, Sinha, P, A, Zhao, H. 2010. A Hybrid Attribute Selection Approach for Text Classification. *Journal of the Association for Information Systems*; 491-518.
- Hall, M, A, Smith, L, A. Feature subset selection: a correlation based filter approach. *Proceedings of the 1997 International Conference on Neural Information, New Zealand, 1997*, pp 237-241.
- Carnegie Mellon University, Pittsburgh. <http://boston.lti.cs.cmu.edu/classes/95-65/HW/HW2/> (accessed at 10.02.2018).
- Dri, A, Abran, A, Mbarki, S. An Experiment on the Design of Radial Basis Function Neural Networks. *International Conference on Information & Communication Technologies, 2006*, pp 1612-1617.