BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 4
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0037

Robust Email Spam Filtering Using a Hybrid of Grey Wolf Optimiser and Naive Bayes Classifier

Jamal Zraqou[1], *Adnan H. Al-Helali*[2], *Waleed Maqableh*[3], *Hussam Fakhouri*[4], *Wesam Alkhadour*[5]

[1]*Department of Virtual & Augmented Reality, University of Petra, Amman, Jordan*
[2]*Department of Cyber Security, Irbid National University, Irbid, Jordan*
[3]*School of Creative Media, Luminus Technical University College, Amman, Jordan*
[4]*Department of Data Science & AI, University of Petra, Amman, Jordan*
[5]*Department of Civil Engineering, Isra University, Amman, Jordan*
E-mails: *Jamal.Zraqou@uop.edu.jo Adnan_Hadi@inu.edu.jo W.Maqableh@saejordan.com Hussam.Fakhouri@uop.edu.jo Wesam.Alkhadour@iu.edu.jo*

Abstract: *Effective spam filtering plays a crucial role in enhancing user experience by sparing them from unwanted messages. This imperative underscores the importance of safeguarding email systems, prompting scholars across diverse fields to delve deeper into this subject. The primary objective of this research is to mitigate the disruptive effects of spam on email usage by introducing improved security measures compared to existing methods. This goal can be accomplished through the development of a novel spam filtering technique designed to prevent spam from infiltrating users' inboxes. Consequently, a hybrid filtering approach that combines an information gain philter and a Wrapper Grey Wolf Optimizer feature selection algorithm with a Naive Bayes Classifier, is proposed, denoted as GWO-NBC. This research is rigorously tested using the WEKA software and the SPAMBASE dataset. Thorough performance evaluations demonstrated that the proposed approach surpasses existing solutions in terms of both security and accuracy.*

1. Introduction

Email or electronic mail is a cheap, efficient, and fast way to exchange messages over the Internet. It is a preferred formal and informal means of communication that is being used by nearly 2.3 billion people worldwide. By the end of 2020, more than five billion users worldwide will have an e-mail account [1]. With the increasing reliance on email as a means of communication, various problems have arisen in the form of spam emails due to the exploitation of the service for illegal purposes [2]. Filtering these emails requires proper classification and research, which is considered a difficult task [3].

There are many ways to deal with spam emails. Filtering is a very popular technique. This involves selecting spam emails from legitimate emails and removing

them. A big problem with spam filters is that they can sometimes exclude legitimate emails, which is called a false positive. When measuring the efficiency of spam filters, the ability of the filter to receive fewer false positives and true negatives is measured [3].

Rule-based spam filtering, content-based spam filtering, and statistical spam filtering are examples of the most common spam filtering techniques.

The rule-based filtering can further be subdivided into whitelist, blacklist, and grey list categories, while content-based is further sub-classified into Keyword list and Distributed checksum filters.

Recently, methods of machine learning or statistical spam filtering have been implemented, which are believed to be more efficient than the knowledge engineering approach. No rules are required, but a set of training samples that are pre-classified e-mails are needed. Then a specific machine learning algorithm is used to find out the classification rules from these spam e-mails. Numerous academicians and researchers techniques have been conducted on machine learning and many of them were applied in the field of spam e-mail classification. Examples of these classification methods include Clustering, Naïve Bayes (NB), Firefly Algorithm, Rough Set, Neural Networks (NN), Support Vector Machines (SVM), Decision Trees (NB Tree, C4.5/J48, and Logistic Model Trees), Ensemble, Random Forests, and Deep Learning.

The Grey Wolf Optimizer (GWO) Algorithm has been developed to handle and solve optimization problems [4]. It is a physically inspired algorithm that mimics a natural or universal phenomenon of the black hole. The advantage of the GWO Algorithm over the other algorithms is shown in its simple structure, and it is a parameter-free algorithm. One of the extensions of the GWO Algorithm is called Binary Grey Wolf Optimizer (BGWO).

The main objective of this research is to limit the effect of spam on the e-mail system work, from both security and economic sides. This would be done through the design of a new spam filtering technique that keeps the spam out of the mailbox of the users. The proposed filtering technique is designed to be a hybrid filter and GWO wrapper feature selection algorithms with the Naive Bayes Classifier (NBC) named by GWO-NBC. The proposed technique has been implemented and validated using WEKA software, Microsoft C#.net, and SPAMBASE dataset downloaded from UCI machine learning repositories. Many performance measurements have been carried out with GWO-NBC algorithms. Experimental results reveal enhancements in security, efficiency, and accuracy compared with the spam detection techniques that are mentioned in this paper. This paper is organized into five sections in addition to Section 1. Section 2 states the related works. Section 3 explores a description of the fundamental method algorithms, while Section 4 presents the proposed spam filtering process and various parts of this technique. Section 5 lists the experiment results and their evaluation. Finally, Section 6 concludes the study.

2. Related works

Many researchers and academicians have developed and reported a variety of e-mail spam classification techniques. A good review and comparison of such filters are presented by Emmanuel Dada [5]. This study introduces the important concepts, attempts, capability, and research direction of machine learning spam filtering. In [6], a global optimization technique using algorithm ABFPA has been used to extract spam detection features. Their method gives a very low performance compared with some widely used techniques. Akshita Tyagi [7] has implemented a deep learning Java (DL4J) method to apply a spam filter. This technology uses the Enron, PU3, PU2, PU1, and PUA e-mail spam datasets. The accuracy, recall, and F1 performance of the proposed technology are compared with DBN, SDAE, and Dense MLP. The main weakness of the DBN and SDAE is the large amount of time consumed in the training phase. In [8], a spam detection system with SVM, PSO, and ANN has been implemented. The presented method is compared with other machine learning methods such as SVM, SOM, PSO, and KNN. The presented technique has low performance. Measurements in evaluating system performance such as false positives, computation time, accuracy, and recall are not used. HC-RBFPSO algorithms for classifying spam have been introduced in [9]. This method is compared with PSO, RBFNN, MLP, and ANN. This method could effectively act as a reliable alternative to other current spam classification techniques. Only accuracy performance has been evaluated. Other parameters such as false positives, computation time, and recall are not taken into account in evaluating the system. The proposed technique does not present an improvement over other latest methods.

A multi-stage NN to the e-mail spam filter has been introduced in [10]. The proposed method conducts multilayer perceptron and classification. The proposed method has produced a better result in spam filtering compared to its application to the classification of scenes in multi-spectral images, where it has been originally implemented. The performance of the algorithm for spam filtering is poor. The time taken to train the dataset in the proposed method is very long. In [11], the hybrid of Spearmen Correlation with KNN Classification as a new technique for e-mail spam filtering is illustrated. The performance of the proposed method has been evaluated in terms of accuracy, precision, recall, and F-measure and compared with other classification methods such as Spearman with KNN and Euclidean with KNN. The performance of the proposed spam filtering method was very poor compared with existing methods. Also, the hybrid of NSA and PSO using LOF as the fitness function for e-mail classification is presented in [12]. Only accuracy performance has been evaluated. Other metrics such as false positives, computation time, accuracy, and recall have not been taken into account in evaluating the system. The proposed work needs to be further optimized to improve its efficiency. In [13], a hybridized ACO and SVM algorithm for e-mail spam classification is proposed. This method performs better than some well-known classification techniques (such as SVM, NB, and KNN) in terms of precision, accuracy, and recall. The adoption of the ACO algorithm for feature selection provides better efficiency in classifying spam messages. The disadvantage of the proposed technique is its low performance.

3. Method algorithms

- Feature selection algorithms

Feature selection algorithms help the classifiers by determining the most relevant subset of features, which enhances classification accuracy and decreases the time required for the training process. The proposed feature selection in this paper consists of two main parts. The first part is the filter method (i.e., IG), while the second part is the GWO Algorithm. Both parts are explained as follows.

3.1. Information Gain (IG)

Information gain is used to measure the number of bits of information obtained for class prediction by knowing the presence or absence of a word in the document. The random event x is the occurrence of the word w and the possible states are the two categories spam and legitimate. Let $c_j \in$ {spam, ham} indicate the set of categories ($K = 2$). The next equation presents the IG for each word in the training set [14],

$$\mathrm{IG}(w) = -\sum_{j=1}^{K} P(c_j)\log P(c_j) + P(w)\sum_{j=1}^{K} P(c_j \mid w)\log P(c_j \mid w) + P(\overline{w})\sum_{j=1}^{K} P(c_j \mid \overline{w})\log P(c_j \mid \overline{w}),$$

where: IG is calculated for each word in the training set;

$P(w)$ is calculated from the number of documents in which the word $\boldsymbol{w}$ occurs divided by the total number of documents N;

$P(c_j)$ is calculated from the number of documents in the training set that belongs to a class c_j divided by N;

$P(c_j/w)$ is calculated from the number of documents c_j that has at least one appearance of w;

$P(c_j/\overline{w})$ is calculated from the number of documents c_j that do not contain w.

3.2. Grey Wolf Optimizer (GWO) Algorithm

In [4], a new meta-heuristic method called Gray Wolf's Algorithm was developed. This algorithm is used for optimization problems and has been modelled mathematically. GWO Algorithm is a population-based algorithm, consisting of many wolves; each wolf has several dimensions based on the nature of the optimization problem itself. In this study, the number of dimensions equals the number of features after applying an IG filter. The mechanisms of social hierarchy, prey encirclement, hunting, attack, and prey hunting are presented in the following sections:

3.2.1. Social hierarchy

The first and highest level in the hierarchy and the packet leader is called alpha (α). Alpha can be a male or female wolf. The alpha (α) position depends on the strength and combat ability. The entire pack is subject to the decision dictated by the alpha. Beta (β) wolves occupied the second level in the hierarchy. Beta acts as an advisor to Alpha in the decision-making. Beta also orders lower-level wolves and keeps discipline up the pack. Delta (δ) wolves rank the third in the grey wolf's social hierarchy. They take orders from alpha and beta wolves. Deltas are the aged wolves that monitor the borders of the pack's territory and warning the pack if there is any

danger. They also play the role of caregivers and patients wounded Wolves. Omega (ω) wolves are at the lowest ranking level of the social population hierarchy. Omega wolves obey the orders of all the dominant wolves and are allowed to eat after everyone has eaten. Fig. 1 shows the whole ranking levels of the GW population hierarchy.

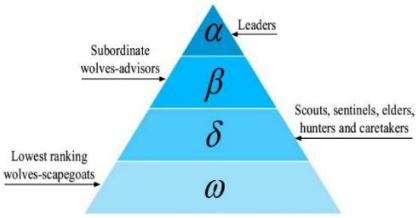

Fig. 1. The Ranking levels of the Population hierarchy [4]

3.2.2. Encircling prey

The encircling behaviour of gray wolves can be modelled mathematically with the next equations,

(1) $$D_{ij} = \left| C_{ij} . \mathrm{XP}_j(t) - X_{ij}(t) \right|,$$

(2) $$X_{ij}(t+1) = \mathrm{XP}_j(t) - A_{ij} . D_{ij},$$

where: A_{ij} and C_{ij} vectors are the randomization coefficients for the *j*-th element of *i*-th wolves, *t* denotes the current iteration, XP_j indicates the location vector of prey, and X_{ij} indicates the location of a grey wolf. A_{ij} and C_{ij} vectors are calculated by the next equations:

(3) $$A_{ij} = 2 * a * \mathrm{rand}[0,1] - a,$$

(4) $$C_{ij} = 2 * a * \mathrm{rand}[0,1].$$

The value of *a* is linearly decreased from 2 to 0 during the iterations. The next equation is used to update the parameter *a*,

(5) $$a = 2 - (2 * t/t_{\mathrm{MAX}}),$$

where *t* shows the current iteration and t_{MAX} is the maximum number of iterations.

3.2.3. Hunting

Prey hunting is guided by a wolf α followed by wolves β and δ, while wolves ω update their locations according to them. Mathematically, prey hunting can be illustrated by the next three equations:

(6) $$\mathrm{D1_{ij}} = \left|\mathrm{C1_{ij}}.\mathrm{X\alpha_j} - \mathrm{X_{ij}}\right|,\ \mathrm{D2_{ij}} = \left|\mathrm{C2_{ij}}.\mathrm{X\beta_j} - \mathrm{X_{ij}}\right|,\ \mathrm{D3_{ij}} = \left|\mathrm{C3_{ij}}.\mathrm{X}\delta_{\mathrm{j}} - \mathrm{X_{ij}}\right|,$$

(7) $$\mathrm{X1_{ij}} = \mathrm{X\alpha_j} - \mathrm{A1_{ij}}.(\mathrm{D1_{ij}}), \mathrm{X2_{ij}} = \mathrm{X\beta_j} - \mathrm{A2_{ij}}.(\mathrm{D2_{ij}}), \mathrm{X3_{ij}} = \mathrm{X}\delta_{\mathrm{j}} - \mathrm{A3_{ij}}.(\mathrm{D3_{ij}}),$$

(8) $$X_{ij}(t+1) = (X1_{ij} + X2_{ij} + X3_{ij})/3.$$

3.2.4. Attacking and searching prey

Vector **A** is used to determine whether the omega wolves are attacking or searching for prey. When |**A**|<1 wolf attack the prey, and when |**A**| > 1wolves search for prey.

The value of **A** decreases linearly from 2 to 0, so the value of the vector **A** falls in the range [–2a, 2a]. The Pseudocode of the GWO Algorithm is illustrated in Fig. 2.

GWO Algorithm

1. **Input:** Fset X_{ij} (j=1,2,...,57), Max ltr = 50, ltr = 1
2. **Output:** Best Solution Xα
3. **Procedure:**
4. Define Function f (X*ij*)
5. Set the size of wolves' pack population, Xij (i=1, 2… 100)
6. Initialize each wolf position in the population randomly, by the following equation:
   Xij = (Upper-Lower) * rand (0,1) + Lower
7. Initialize the vector's a, Aij and Cij
6. Calculate the Finesses value for each grey wolf in population
7. Sort and rank the entire pack in ascending order based on fitness values
7. Set the three lowest costs: Xα as 1st best wolf, Xβ as the 2nd best wolf, and Xδ as 3rd best wolfs
8. Set 'a' = 2
8. While ($itr \leq MaxItr$) do
9. For (each grey wolf) do
10. Renew the position of the current wolf Xα, X_β ,and X_δ
11. End For
12. Evaluate the Finesses value for all grey wolfs in population
13. Update the best three grey wolfs Xα, X_β and X_δ
14. Decrease 'a' uniformly using equation a = 2 – (2 * t / t_MAX)
15 Update a, Aij, and Cij
16. $itr = itr + 1$
17. End while
18. Return the best solution of Xα

Fig. 2. The Pseudocode of GWO Algorithm

3.3. The Naïve Bayesian Algorithm

The Naïve Bayes Algorithm is a simple probabilistic classifier that computes a set of probabilities by calculating the frequency and combining the values into a given data set. NBC was designed with the fact that there is no relationship between the presence or absence of a particular feature and the presence or absence of any other feature. According to this, the specific nature of the probability model, NBC can be efficiently used in supervised learning [15].

A classification problem is represented by the vector **F** = (F_1, F_2, F_3, ..., F_N), where *n* represents the number of independent features. Then the instance probabilities of this condition can be calculated by p (C_k| F_1, F_2, and F_3, …, F_N). The conditional probability of Bayes' theorem is represented by the next equation [18]:

$$p\,(C_k \mid \mathbf{F}) = \frac{p(C_k).p(\mathbf{F} \mid C_k)}{p(\mathbf{F})}. \tag{9}$$

This equation can be written as posterior = prior × likelihood / evidence. This means that the denominator P(**F**) does not depend on C_k and the values of the features so that the P(**F**) is constant and the conditional distribution over the class variable can be calculated by the next equation:

$$p\,(C_k \mid F_1, F_2, \ldots., F_n) = \frac{1}{z}\,p\,(C_k) \prod_{i=1}^{n} p(F_i \mid C_k)\,, \tag{10}$$

where *z* is the evidence scaling factor and it depends only on (F_1, F_2, F_3, …., F_N), which means that it is a constant if the values of the features are known.

4. The proposed spam filtering process

The proposed spam filtering process is conducted in three phases. Phase 1: Pre-processing (Normalization); Phase 2: Feature selection (IG-GWO); Phase 3: Classifier (NBC). Fig. 3 illustrates the block-diagram of the proposed filtering system.

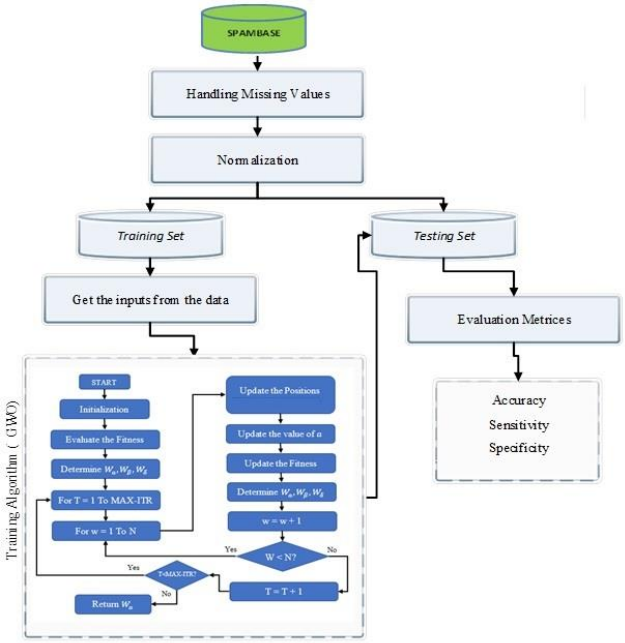

Fig. 3. The block-diagram for the proposed filtering system

5. Experimentation results and evaluation

5.1. Settings

The assessments of the proposed algorithm were carried out on a laptop personal computer (CPU/ Intel Core™ i5, 2.5 GHz, RAM/ 8 GB, Operating System: Windows 10 / 64-bit). The NBC Algorithm is imported from the WEKA class file. WEKA is a popular set of machine learning algorithms written in Java for data mining and analysis tasks. WEKA is an open-source machine learning tool, consists of tens of models, which can be adapted to any other programming language. The IG Algorithm is also imported from WEKA, and modified to produce the values of the weights from the original features. The BGWO Algorithm has been developed and written using Microsoft C#.net with Visual Studio.net 2019.

5.2. Evaluation based on SPAMBASE datasets

The SPAMBASE dataset used for evaluating the proposed filtering system is downloaded from [16], which is a very popular Spam e-mails dataset and has been heavily used in the ML research area. It consists of 4601 instances (e-mails) with 195

samples; each e-mail consisted of 57 attributes representing word frequencies. The e-mails are divided into two main classes; Spam (1813 ≅ 39%) and Ham (2788 ≅ 61%). The attributes of the SPAMBASE dataset are given in Table 1.

Table 1. The SPAMBASE attributes [16]

Attr. No	Attribute type	Attribute description
1-48	word_freq_WORD	The percentage of words in the e-mail that match WORD
49-54	char_freq_CHAR	The percentages of characters in the e-mail that match CHAR
55	capital_run_length_average	The average length of continuous capitalization sequences
56	capital_run_length_longest	The length of the longest sequence of uppercase characters without interruption
57	capital_run_length_total	The total uninterrupted sequence length of uppercase = The total number of uppercase characters in the e-mail
58	classes attribute	Indicates whether the e-mail is considered spam (1) or not (0)

5.3. Evaluation matrices

Equations (12) and (13) are used for measuring the classification Accuracy (Acc), Error Rate (Err = 1 − Acc), Spam Recall (SR), Spam Precision (SP), and weighted average precision and recall (F-measure). Table 2 represents the classification types in the proposed filtering system, where:

- TP is truly positive;
- TN is a true negative;
- FP is a false positive;
- FN is a false negative.

Table 2. The classification types

Message	Classified as ham	Classified as spam
Ham message	TP	FP
Spam message	TN	FN

(11) $$\text{Acc} = \frac{\text{TP+FN}}{\text{TP+FP+TN+FN}} \quad \&$$

$$\text{Err} = \frac{\text{FP+TN}}{\text{TP+FP+TN+FN}},$$

(12) $$\text{SR} = \frac{\text{FN}}{\text{FN+TN}}, \quad \text{SP} = \frac{\text{FN}}{\text{FN+FP}}, \quad \&$$

$$\text{F-measure} = 2 * \frac{\text{SR}}{\text{SR+SP}}.$$

5.4. Results and discussion

5.4.1. Results of pre-processing

The SPAMBASE dataset samples are normalized using the MinMax normalization function. The normalization process is used to decrease the variation between the features, which leads to a decrease in the noise in the dataset. MinMax method can be calculated by the equation

(13) $$f_i^{\text{new}} = \frac{f_i^{\text{old}} - \min(F)}{\max(F) - \min(F)},$$

where f_i represents the current value of the feature, max (f) and min (f) denote the max and min. values of the feature f, respectively. All these features are sent to the Information Gain (IG).

5.4.2. Results of information gain

The information gain is a filter method used to calculate the weights of all features. The results of this stage are displayed in Table 3.

According to the backward feature selection method, the features with weights lower than the threshold value are removed, while the rest of the features are kept for the next stage (GWO Algorithm). The value of the threshold is determined using by Trial & Error method, which is equal to 0.06.

Table 3. Results of information gain for all features

F	Value	F	Value	F	Value	F	Value	F	Value
1	0.0633940914	13	0.0587703815	25	0.1519977816	37	0.0861659930	49	0.0174822925
2	0.1152055475	14	0.0464861759	26	0.1713142710	38	0.0468924491	50	0.0293570741
3	0.0766860962	15	0.1484891557	27	0.1667884601	39	0.0586612127	51	0.0256490915
4	0.0989203670	16	0.2088795931	28	0.0967945596	40	0.0239568790	52	0.1468950830
5	0.0968304546	17	0.1054739180	29	0.1427346434	41	0.1071717166	53	0.1997369216
6	0.0783175396	18	0.0826097454	30	0.1259291588	42	0.0966195662	54	0.0770728390
7	0.3183780340	19	0.063568940t5	31	0.1260543369	43	0.0625601933	55	0.0837693808
8	0.1242538724	20	0.1923898936	32	0.1136429636	44	0.0700218131	56	0.0875935402
9	0.1006605799	21	0.1322292620	33	0.0510928448	45	0.0366325652	57	0.0665749107
10	0.0763677165	22	0.0856459108	34	0.1016220546	46	0.0931757605		
11	0.1565211746	23	0.1993059586	35	0.1150374635	47	0.0767729685		
12	0.0385116955	24	0.2743964881	36	0.0420063978	48	0.0655275554		

5.4.3. Results of GWO

Each experiment was executed 10 times using a different number of iterations and number of agents (wolves pack) in each time and the results are listed in Table 4. It shows the accuracies of all runs (10 times) for (100, and 300) iterations and a specific number of agents (10, 30, and 50). Moreover, the table also presents the average for all accuracies.

In general, the proposed algorithm showed a huge improvement over the original accuracy of the NBC algorithm, as the GWO Algorithm supported NBC to select the most relevant features. The original accuracy for NBC based on all 57 features was around (79.41%), while the worst result achieved with the support of GWO was around (96.390 %).

Moreover, Table 4 illustrates that the number of agents has a great impact on the searching process of the GWO Algorithm, as the number of agents increased, the value of classification accuracy increased as well. The main reason behind this fact, that the possibility of finding better results is bigger when more agents are utilized for the searching process. For example, if the population size equals 50 agents, this means that 50 possible solutions are trying to reach better positions in a single iteration. On the other hand, the table also showed that the number of iterations affects

reaching the best solutions, as the number of iterations increased, the algorithm reached better classification accuracies.

Table 4 summarizes the results for all runs and experiments. It presents the best, worst, average, and standard deviation for all run times. Moreover, it presents the results in terms of the number of features. Table 5 presents the other evaluation metrics, which are precision, recall, and F-measure. It can be judged that the algorithm is stable and produces high precision and recall values.

Table 4. The results of the proposed algorithm for different agents and iterations.

Number of wolves	Iterations	Statistic	Accuracy (%)	Features	Error rate	Binary representation
10	100	Best	96.95525	28	3.044750	01011010101011011111110010001001110011011111
		Worst	96.39016	30	3.609845	00110111110111111111101100011000011101011111110
		Mean	96.71834			
		Std	0.314336			
	300	Best	97.95503	30	2.044967	11111111111010101101010110100010001110111111
		Worst	97.04219	27	2.957812	11111111011110101011111000000000111010101010101
		Mean	97.43993			
		Std	0.407387			
30	100	Best	97.43341	25	2.566593	01100010001111011011110000100001110011011111
		Worst	97.15086	22	2.849140	01110010101010111011101110000010000010101010110
		Mean	97.32691			
		Std	0.202979			
	300	Best	98.04197	25	1.958030	10111110101010001111111100010000000110101111110
		Worst	97.78116	29	2.218843	11011111111011111111110100000101000110011010101
		Mean	97.93547			
		Std	0.202979			
50	100	Best	98.19411	24	1.805889	10111010010110011111110101100000010010001111110
		Worst	97.91156	24	2.088436	00110111101111111101110110010001110000111011010
		Mean	98.03762			
		Std	0.223514			
	300	Best	98.95482	27	1.045185	00101111111011011011110100000101010011011111110
		Worst	98.45492	25	1.545076	01010111011110111111110101000100000011011010010
		Mean	98.68531			
		Std	0.284870			

Table 5. Accuracy, precision, recall, and F-measure for proposed model

Stars	Iterations	Features	Accuracy	Spam precision	Spam recall	F-measure
–	–	57	79.41751	84.400	79.600	**79.800**
10	100	28	96.95525	97.105	97.205	**97.105**
	300	30	97.95503	97.405	97.405	**97.405**
30	100	25	97.43341	96.505	96.505	**96.505**
	300	25	98.04197	97.805	97.605	**97.605**
50	100	24	98.19411	97.305	97.405	**97.305**
	300	27	98.95482	98.905	98.905	**98.905**

5.4.4. The proposed method and evaluation

The achieved accuracy and error rate results for the proposed GWO-NBC approach are compared with the most widely used spam filtering approaches, namely NBC,

SVM, KNN, ACO-SVM, GA-NBC, ACO-NBC, PSO-NSA, and DFS [13, 17, 19], and plotted in Fig 4.

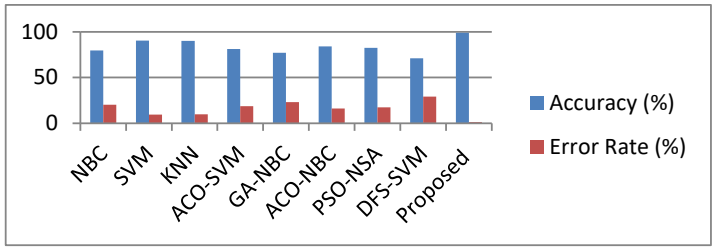

Fig. 4. Comparisons in terms of accuracy and error rate for different ML filters

It should be noted that assessments of all these approaches have used the same SPAMBASE dataset. GWO-NBC approach has achieved higher accuracy and lower error rate than the former approaches.

6. Conclusion

A new hybrid spam filtering approach has been proposed to achieve effective and efficient spam detection by identifying optimal features. The proposed approach combines a GWO optimization algorithm and an NBC classifier to determine the optimum feature. The new method has been tested on the "SPAMBASE" dataset from UCI machine learning repositories.

In comparison with other methods, the results obtained from the proposed approach have outperformed the related works mentioned in this paper in terms of accuracy and error rate. The proposed approach has achieved an accuracy of 97.61% which is higher than all the considered approaches. The proposed model was implemented using WEKA software, and Microsoft C#.net, on an Intel Core i5 machine.

***Acknowledgements*:** The authors are very grateful to the University of Petra, Irbid National University, Luminus Technical University College, and Isra University, Amman, Jordan for the financial support granted to cover the publication fee of this research article.

References

1. Gaurav, D., S. Tiwari, A. Goyal, N. Gandhi, A. Abraham. Machine Intelligence-Based Algorithms for Spam Filtering on Document Labeling. – Soft Comput., Vol. **24**, 2020, pp. 9625-9638.
2. Dedeturk, B., B. Akay. Spam Filtering Using a Logistic Regression Model Trained by an Artificial Bee Colony Algorithm. – Applied Soft Computing, Vol. **91**, 2020, 106229. ISSN 1568-4946.
3. Pelletier, L., J. Almhana, V. Choulakian. Adaptive Filtering of SPAM. – White Paper, University of Moncton E1A 3E9, 2018.
   www.umoncton.ca/greti/papers/ Adaptive Filtering of Spam.pdf
4. Mirjalili, S., S. M. Mirjalili, A. Lewis. Grey Wolf Optimizer. – Advances in Engineering Software, Vol. **69**, 2014, pp. 46-61.

5. D a d a, E. G., J. S. B a s s i, H. C h i r o m a, S. M. A b d u l h a m i d, A. O. A d e t u n m b i, O. E A j i b u w a. Machine Learning for Email Spam Filtering: Review, Approaches and Open Research Problems. – Heliyon, Vol. **5**, 2019, Issue 6, e01802. ISSN 2405-8440.
6. R a j a m o h a n a, S., K. U m a m a h e s w a r i, B. A b i r a m i. Adaptive Binary Flower Pollination Algorithm for Feature Selection in Review Spam Detection. – In: Proc. of IEEE International Conference on Innovations in Green Energy and Healthcare Technologies, 2017, pp. 1-4.
7. T y a g i, A. Content-Based Spam Classification – A Deep Learning Approach. A Thesis Submitted to the Faculty of Graduate Studies. University of Calgary, Alberta, Canada, 2016.
8. Z a v v a r, M., M. R e z a e i, S. G a r a v a n d. E-mail Spam Detection Using Combination of Particle Swarm Optimization and Artificial Neural Network and Support Vector Machine. – Int. J. Mod. Educ. Comput. Sci., 2016, pp. 68-74.
9. A w a d, M., M. F o q a h a. E-mail Spam Classification Using Hybrid Approach of RBF Neural Network and Particle Swarm Optimization. – Int. J. Netw. Secur. Appl., Vol. **8**, 2016, No 4.
10. A l k a h t, I., B. A l-K h a t i b. Filtering SPAM Using Several Stages Neural Networks. – Int. Rev. Comp. Software, Vol. **11**, 2016, No 2.
11. S h a r m a, A., A. S u r y a w a n s i. A Novel Method for Detecting Spam E-mail Using KNN Classification with Spearman Correlation as Distance Measure. – Int. J. Comput. Appl., Vol. **136**, 2016, No 6, pp. 28-34.
12. P a l a n i s a m y, C., T. K u m a r e s a n, S. E. V a r a l a k s h m i. Combined Techniques for Detecting E-mail Spam Using Negative Selection and Particle Swarm Optimization. – Int. J. Adv. Res. Trends Eng. Technol., 2016. ISSN: 2394-3777.
13. K a r t h i k a, D., P. V i s a l a k s h i, T. S a n k a r. Improving E-mail Spam Classification Using Ant Colony Optimization Algorithm. – Int. J. Comput. Appl., 2015, No ICICT, pp. 975-8887.
14. G o e t s c h i, R. SPAM-Filtering Using Artificial Neural Networks. Semester Thesis, Berne University of Applied Sciences, July 2004.
www.hta-bi.bfh.ch/~goetr/nn_spam_goetschi2004.pdf
15. S m i t a, M., S. K u m a r. Survey on Types of Bug Reports and General Classification Techniques in Data Mining. – International Journal of Computer Science and Information Technologies, Vol. **6**, 2015, No 4, pp. 1578-1583.
16. UCI Machine Learning Repository Spambase Dataset. The University of California, School of Information and Computer Science.
http://archive.ics.uci.edu/ml/datasets
17. I d r i s, I., et. al. A Combined Negative Selection Algorithm-Particle Swarm Optimization for an E-mail Spam Detection System. – Eng. Appl. Artificial Intelligence, Vol. **39**, 2015, pp. 33-44.
18. L a m i a a, M., E. B a k r a w y. Grey Wolf Optimization and Naive Bayes Classifier Incorporation for Heart Disease Diagnosis. – Aust. J. Basic & Appl. Sci., Vol. **11**, 2017, No 7, pp. 64-70.
19. U y s a l, A. K., S. G u n a l. A Novel Probabilistic Feature Selection Method for Text Classification. Knowledge-Based Syst., Vol. **36**, 2012, pp. 226-235.

Received: 07.09.2023; Second Version: 20.09.2023; Accepted: 29.09.2023 (fast track)