Adv Syst Sci Appl. 2019; **03**;11–22
Published online at https://ijassa.ipu.ru/index.php/ijassa/article/view/751

Hybrid Particle Swarm Optimization and Pegasos Algorithm for Spam Email Detection

Lamiaa M. El Bakrawy*

Faculty of Science, Al-Azhar University, Cairo, Egypt
E-mail: dr_lamiaa_el_bakrawy@azhar.edu.eg
Received May 25, 2019; Revised September 9, 2019; Published October 1, 2019

Abstract: Email is one of the most popular communication tools for most internet users nowadays. It has become fast and an effective method to share and exchange information all over the world. Despite the great advantages of emails, its usage is facing problem which is spam emails. Spam emails are the huge presence of bulk and unsolicited emails which are expensive for the companies, consume a huge amount of mail servers, network bandwidth and waste of time. Isolating and detecting these emails is known as spam detection. Many spam detection methods have been proposed but there is still need to detect the email spam effectively with high accuracy. In this paper, hybrid particle swarm optimization and Pegasos algorithm, which is called (PSO-Pegasos) is proposed for spam email detection. Particle swarm optimization is employed as a search strategy to determine the optimal parameters for Pegasos algorithm in order to achieve higher performance. The proposed algorithm has been applied on spambase dataset downloaded from UCI Machine Learning Repository. Experimental results demonstrate that the proposed algorithm outperforms the performance of all the earlier proposed algorithms, considering the accuracy, recall, precision and F-measure on the same dataset.

Keywords: Particle swarm optimization, Pegasos algorithm, Email spam, Spam detection, Accuracy.

1. INTRODUCTION

Nowadays, email is one of the most popular communication tools for most internet users because of its free availability and efficiency [1, 2]. Email is a method of receiving and sending information over electronic networks such as the internet. However, the major problem is the presence of bulk and unsolicited email which is known as spam. Spammer is the person who sends mass quantity of spam emails and collects email addresses from chatrooms, viruses, customer lists and websites. Spam email consumes a huge amount of mail servers, network bandwidth and wastes users' time to remove all spam emails which causes lower productivity. Thus, how to isolate and detect spam email in efficient way with high accuracy becomes an important study.

Spam email detection can be considered as classification problem which is used to detect the spam emails one by one to classify email as spam or non-spam [3]. In recent years, most of the spam detection algorithms based on machine learning techniques is used, but still the reported accuracy requires more work to accomplish better accuracy. Sabri et al. in [4] presented continuous learning approach based on artificial neural network (CLA_ANN) for spam email detection. They made core modifications in the input layer of artificial neural network to substitute the useless layers with new favorable layers and to be varied with time.

*Corresponding author: dr_lamiaa_el_bakrawy@azhar.edu.eg

The results showed that applying CLA_ANN using 300 input layers succeeded in achieving 3.668 % false negative and 0.534 % false positive. Zhang et al. in [5] presented naive bayes model for spam email detection by applying cost-sensitive multi-objective genetic programming for feature extraction and achieved an accuracy of 79.3%. Renuka et al. in [6] proposed spam classification algorithm using hybrid ant colony optimization and naive bayes classifier and applied it on spambase dataset. The accuracy obtained was 84% which indicated that the hybrid algorithm outperformed hybrid genetic algorithm and naive bayes tested on the same dataset.

Özgür et al. in [7] used artificial neural network and bayesian filter for spam email detection. They considered two artificial neural network structures, multi layer perceptron and single layer and the inputs are specified based on probabilistic and binary models. Experimental results for 750 e-mails (410 spams and 340 non-spam), achieved 90% accuracy. Temitayo et al. in [8] used genetic algorithm to optimize the support vector machines (SVM) classification parameters. The hybrid algorithm achieved 90% accuracy for the testing set. Liu et al. in [9] proposed a new learning method (PSO-LM) for process propagation neural networks (PNNs) based on particle swarm optimization (PSO) and gaussian mixture functions. Experiments results showed that applying (PSO-LM) on spambase dataset achieved 90.5% accuracy for the testing set which is better than back propagation neural networks (BPNNs) and basis function expansion based learning method (BFE-LM). Moreover, Idris and Selamat in [10] presented a hybrid model of negative selection algorithm (NSA) and particle swarm optimization (PSO). They worked on spambase dataset and achieved 91.22% accuracy for the testing set.

Awad and Foqaha in [1] proposed a hybrid algorithm of rbf neural network and particle swarm optimization (HC-RBFPSO) for spam email classification. They used particle swarm optimization algorithm to optimize the parameters of Radial Basis Function Neural Networks (RBFNN) based on the evolutionary heuristic search of PSO. They divided spambase dataset into 70% training set and 30% testing set. Experiments are measured by using a different number of hidden layer starting from 10 to 50. The accuracy obtained was 91.4% for the testing set which was concluded that the hybrid approach had good performance compared to other algorithms tested on the same dataset. Olatunji in [11] proposed support vector machines-based model for spam detection. He used a systematic parameter search in order to achieve better spam detection accuracy. The accuracy obtained was 94.06% for the testing set. Experimental results show that the proposed scheme outperformed other published algorithms tested on spambase dataset used in this work.

Considering the performance accuracy achieved till now, there is still need to try to achieve better results on the same dataset. The main aim of this paper is to propose an alternative algorithm that can accomplish a performance higher than previous algorithms. In this paper, Hybrid particle swarm optimization and Pegasos algorithm (PSO-Pegasos) is proposed to achieve better accuracy of spam email detection. Pegasos algorithm is applied to solve the optimization problem cast by support vector machines (SVM) while particle swarm optimization is used as a search strategy to select the optimal parameters (the weights) for Pegasos algorithm, which means in each iteration of PSO, the weights (w-parameters) are changed based on the fitness function (mean squared error). After running PSO algorithm a number of iterations, it will obtain the best optimal w-parameter for Pegasos algorithm. In this paper, comparison of performance measures of Pegasos and hybrid algorithm (PSO-Pegasos) for training and testing sets is presented for spam email detection.

The rest of this paper is structured as follows: The fundamentals of particle swarm optimization and the principles of the original Pegasos algorithm are introduced in Section

2. Section 3 describes the details of the proposed algorithm. Experimental results and discussions are demonstrated in section 4. Finally, section 5 concludes the paper.

2. PRELIMINARIES

2.1. Particle Swarm Optimization

Particle swarm optimization (PSO) was invented by Kennedy and Eberhart in 1995 [12]. PSO is a widely used population-based stochastic optimization technique since it has strong global search capability, high convergence speed, high robustness and is conceptually very simple [13, 14, 15]. PSO is still attracted the attention of a lot of researchers over nearly a quarter century. Particle swarm optimization simulates the social behavior among species such as fish schools, bird flocks.

The set of particles represent a population of the possible solutions. In canonical PSO algorithm, particles are initialized with a population to get a random solution. Then, the particles fly iteratively around in d-dimension search space to search the optimal solution, where the proper fitness function can be calculated according to the problem. Each particle is indicated by a row vector $\vec{x}_i$, where i is the index of the particle, and a velocity indicated by $\vec{v}_i$. The best position of the particle (pbest) is indicated by vector $\vec{x}_i^{\#}$, and its j-th dimensional value is $x_{ij}^{\#}$, while the best position among the swarm (gbest) is indicated by a vector $\vec{x}^{*}$, and its j-th dimensional value is x_j^{*}. In each iteration t, the velocity updating formula of particle is calculated by Eq. (2.1) and the position updating formula of particle is determined by the sum of the previous position and the new velocity by Eq. (2.2).

$$v_{ij}(t+1) = \left\{ \begin{array}{l} wv_{ij}(t) + c_1 r_1 (x_{ij}^{\#}(t) - x_{ij}(t)) \\ +c_2 r_2 (x_j^{*}(t) - x_{ij}(t)) \end{array} \right. \tag{2.1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \tag{2.2}$$

where c_1 and c_2 are nonnegative constants called as learning factors, r_1 and r_2 are random numbers uniformly distributed in U(0,1) for the j-th dimension of the i-th particle. w is the inertia weight, which can increase the algorithm search capability and control the process of algorithms searching. Eq. (2.1) makes each particle tends to move across the design space, considering its own experience, which is the memory of its best fitness function value achieved by the particle in the past, and the experience of its most successful particle in the swarm.

In PSO algorithm, the particles tend to search the solutions in the problem space with a range $[-s, s]$ to prevent the particle from flying away out of the search space. If the range $[-s, s]$ is not symmetrical, it will be changed to the corresponding symmetrical range and the maximum velocity during one iteration must be limited on the interval $[-v_{max}, v_{max}]$ given in Eq.(2.3)

$$v_{ij} = sign(v_{ij}) min(\left|v_{ij}\right|, v_{max}). \tag{2.3}$$

Where the value of v_{max} is $p \times s$, with $p \in [0.1, 1]$ but v_{max} is usually selected to be s, i.e. $p = 1$. The termination criterion for iterations will be determined according to whether the maximum number of iterations or minimum fitness function error is reached.

2.2. Pegasos: primal estimated sub-gradient solver for SVM

Pegasos was described and analyzed by Shalev-Shwartz et al. in [16] for solving the optimization problem cast by support vector machine (SVM). It performed a stochastic sub-gradient descent based on the primal objective by chosen step size carefully to improve

convergence [17, 18, 19]. Pegasos has attracted research interest because it has better convergence bounds and robustly convex optimization objective. It uses theory of strongly convex optimization problems and hinge loss instead of the original linear constraints which makes the objective of SVM unconstrained.
Given a binary classification problem with training set $S = (x_i, y_i)$, $(i = 1, \ldots, N)$, where x_i is a d-dimensional feature vector and $y_i = \pm 1$ is the class label. The goal of linear support vector machines is to find a classifier in the following form

$$h(x) = sign(w^T x), \tag{2.4}$$

where w is the weight vector which can be learnt from training set to solve the following optimization problem after number of iterations T.

$$\min_w = \frac{\lambda}{2}\|w\|^2 + \frac{1}{N}\sum_{(x,y)\in S} l(w,(x,y)), \tag{2.5}$$

where $l(w,(x,y)) = \max(0, 1 - y(w,x))$, and $\lambda \geq 0$ is the regularization parameter. In each iteration t , Pegasos algorithm aims to update w by choosing a random training set $A_t \subseteq S$ with size k, where k is the number of training examples used for calculating sub-gradient through the following approximate objective function

$$f(w, A_t) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{k}\sum_{(x,y)\in A_t} l(w,(x,y)). \tag{2.6}$$

The sub-gradient of the approximate objective function $f(w, A_t)$ at w_t is calculated by

$$\nabla_t = \lambda w_t - \frac{1}{|A_t|}\sum_{(x,y)\in A_t^+} yx, \tag{2.7}$$

where A_t^+ is the set of examples when w suffers a non-zero loss. Finally, the sub-gradient is used to update the weight by using a step size of $\eta_t = \frac{1}{|\lambda_t|}$ as

$$w_{t+1} = w_t - \eta_t \nabla_t, \tag{2.8}$$

where η_t is the learning rate. The last vector w_{T+1} is the output of Pegasos algorithm after number of iterations T.

3. THE PROPOSED ALGORITHM

In this section, we describe the proposed PSO-Pegasos algorithm to determine the optimal values of Pegasos parameters as shown in Figure 3.1. The detailed description is as follows:

3.1. Data preprocessing

In this paper, the popular and often used corpus benchmark spambase dataset is utilized to classify email as spam or non-spam. The dataset is available in numeric form and the features are frequencies of various characters and words in emails. The main tasks in preprocessing are transformation, reduction, cleaning, integration and normalization. Normalization is an important pahse to fast the algorithm, convergence and decrease the influence of imbalance in data. In spambase dataset, normalization is done before running PSO-Pegasos algorithm. Each feature of spambase dataset is normalized in the range [0, 1] through the following function

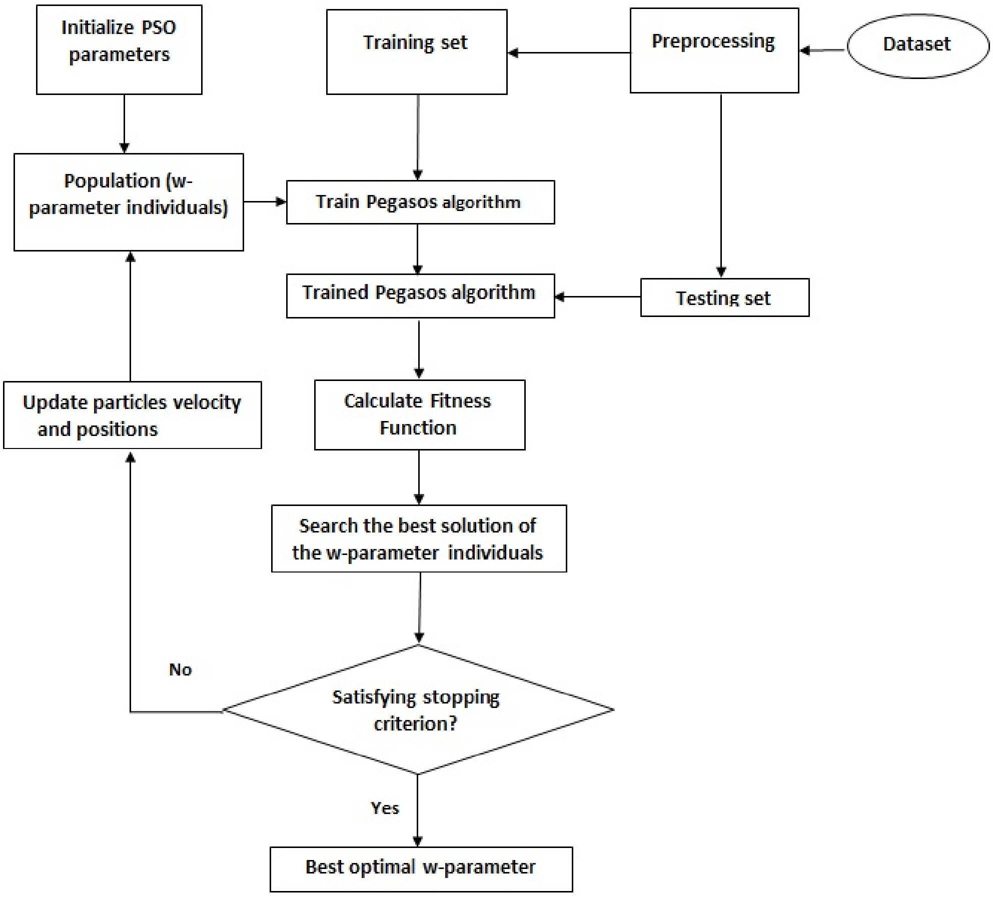
Initialize PSO parameters
Training set
Preprocessing
Dataset
Population (w-parameter individuals)
Train Pegasos algorithm
Trained Pegasos algorithm
Testing set
Update particles velocity and positions
Calculate Fitness Function
Search the best solution of the w-parameter individuals
No
Satisfying stopping criterion?
Yes
Best optimal w-parameter

Figure 3.1: Flowchart of the proposed algorithm.

$$A = \frac{a - min}{max - min} \tag{3.9}$$

where A is the scaled value, a is the original value, max and min are the maximum and minimum bounds of the feature value.

3.2. PSO-Pegasos algorithm

In this research, hybrid particle swarm optimization and Pegasos algorithm (PSO-Pegasos) is proposed for spam email detection. Particle swarm optimization has been utilized to optimize the parameters of Pegasos algorithm. In PSO each solution is called a particle. Fitness function (mean squared error) is used to evaluate the particles for the optimal solution. Particle swarm optimization is used as a search strategy to determine the optimal parameters (weights) for Pegasos algorithm, which means in each iteration of PSO, the weights (w-parameters) are updated depending the fitness function. No assumptions are needed about the w-parameter in Pegasos algorithm since PSO algorithm can help us to identify automatically the best optimal w-parameter (bw) that utilized to obtain the highest classification accuracy for Pegasos algorithm. The major steps of the hybrid Particle swarm optimization and Pegasos algorithm (PSO-Pegasos) are shown as follows:

1. Initialize the population for w-parameter individuals (particles) in a random manner from spambase dataset. Suppose that, each particle swarm position is $X_i = \{a_{i,j}, j = 1, 2, ..., k\}$, where $a_{i,j}$ is j th w-parameter for the i th individual, k is the number of features (attributes) of spambase dataset and the value of the w-parameter for each individual is vector of k random numbers in range from -10 to 10.
2. Initialize velocity of particle swarm optimization randomly in range from -100 to 100
3. Calculate the fitness function for each particle which is acquired by Pegasos algorithm to classify non-spam and spam emails correctly by

$$Fitness = MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2. \tag{3.10}$$

   where MSE is the mean squared error , X is a vector of n predictions, and Y is the vector of true values.
4. If the fitness function is better than the best fitness function of the particle (pbest) then the current position will be (pbest)
5. Select the best position among all particles (gbest) in current iteration
6. Update the velocity of each particle depending on Eq. (2.1).
7. Update the position of each particle (w-parameter) depending on Eq. (2.2).
8. Search the the pbest of particle as (w-parameter) of Pegasos algorithm in same iteration.
9. Repeat steps 3 to 8 until obtaining the best optimal w-parameter (bw) which leads to get the highest accuracy for spam email detection with more exploration in the search space.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this paper, the experiments were performed on a system with a 2.40 GHZ Intel(R) Core(TM)i7 processor and 16 GB memory using written codes in Matlab 15.

4.1. Dataset description

In this work, The dataset utilized is spambase dataset which is used to evaluate the proposed algorithm. Hopkins et al. [20] presented spambase dataset in their colleagues. It has been collected from UCI Machine Learning Repository site. In the spambase dataset, the total email instances is 4601. 1813 from these email instances are characterized as spam (39.4%) and the remaining are non-spam. Spambase dataset consists of 57 features and 1 classification attribute, which is the label of class indicating the status of each email instance whether it is spam (1) or non-spam (0). Most of the features (1-54) show particular characters or words were repeatedly occurring in an email or not. The features from 55 to 57 present the measurement for length of consecutive capital letters. The definitions of the features can be shown as follows

- Features from 1 to 48 are real continuous features which are equal to the percentage of words in the e-mail that match WORD.
- Features from 49 to 54 are real continuous features which are equal to the percentage of characters in the e-mail that match CHAR.
- Feature 55 is real continuous feature which is equal to the average length of continuous sequences of capital letters.
- Feature 56 is an integer continuous feature which is equal to the length of longest continuous sequence of capital letters.
- Feature 57 is an integer continuous feature which is equal to the total number of capital letters in the e-mail.

4.2. Evaluation measures

In this research, the evaluation of the proposed algorithm is carried out based on popular and commonly performance measures such as accuracy, recall, precision, F-measure [21, 22]. The information about these measures is done depending on the confusion matrix presented in Table 4.1.

Table 4.1: Confusion matrix

	Actual class	
	Spam	Non-spam
Predicted class		
Spam	TP	FP
Non-spam	FN	TN

Brief overview of each performance measure is shown below.

- **Accuracy** is defined as the fraction of all emails (non-spam and spam emails) that are classified correctly by the algorithm. It can be represented by the following equation:

$$Accuracy = \frac{TP+TN}{FP+FN+TP+TN} \tag{4.11}$$

  where TP and TN are the number of spam emails and non-spam emails correctly classified, respectively. FP and FN are the number of spam emails and non-spam emails incorrectly classified, respectively.
- **Recall** stands for the proportion of spam emails being recognized and can be represented as follows:

$$Recall = \frac{TP}{FN+TP} \tag{4.12}$$

- **Precision** stands for the fraction of spam emails that are correctly classified as spam.

$$Precision = \frac{TP}{FP+TP} \tag{4.13}$$

- **F-measure** (F-score), denotes the harmonic average of precision and recall and can be written as follows:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.14}$$

4.3. Results and Discussion

The experimental method applied here followed carefully the computational intelligence technique. Spambase dataset was first divided into two phases, training set and testing set in the ratio 7:3, respectively. The data was chosen randomly for training and testing sets in order to exclude any particular behavior of the dataset. Then, the training set (70% of data) was first entered to the algorithm for training and validation and the rest of dataset (30% of data) was used to test the algorithm to ensure the performance accuracy of the proposed algorithm.

To evaluate the proposed algorithm, the parameters settings for original Pegasos algorithm are regularization parameter (λ) with different values from 0.0001 to 0.1 and number of iterations T =1000. The original Pegasos algorithm is applied on spambase dataset with different values of λ and the four performance measures are recorded for training and testing sets as shown in Table 4.2. It can be observed in Table 4.2 that using a small value of λ (0.0001) in a large dataset increases the accuracy, recall, precision and F-measure, respectively for spam detection by Pegasos algorithm. According to this result, we fixed the

Table 4.2: Performance measures for Pegasos algorithm with different values of λ for Spambase dataset

(λ)		Accuracy	Recall	Precision	F-measure
0.0001	training set	93.62 %	0.9360	0.9370	0.9360
	testing set	92.71 %	0.9270	0.9280	0.9270
0.001	training set	93.04 %	0.9300	0.9310	0.9310
	testing set	92.51 %	0.9250	0.9250	0.9250
0.01	training set	92.60 %	0.9260	0.926	0.926
	testing set	91.67 %	0.9170	0.9180	0.9160
0.1	training set	90.43 %	0.9040	0.9060	0.903
	testing set	89.19 %	0.8920	0.8970	0.8900

value of λ as 0.0001 in proposed algorithm (PSO-Pegasos).

In PSO-Pegasos, the parameters of particle swarm optimization were set as learning factors c_1 = c_2 = 1.4, v_{max} = 4 and inertia weight (w) was linearly decreased from 0.9 to 0.4. The population size was fixed to 20 particles to reduce the computational cost and fast the convergence process of the algorithm. Tuning the parameters for particle swarm optimization is important in designing the algorithm. Figure 4.2 shows the effect of the number of iterations on the accuracy of the proposed algorithm PSO-Pegasos using different number of iterations from 5 to 30. As shown in Fig. 4.2, we can observe that when the number of iterations was increased, the accuracy was increased until it accomplished an extent (number of iterations =20) at which increasing the number of iterations did not affect the accuracy of the proposed algorithm.

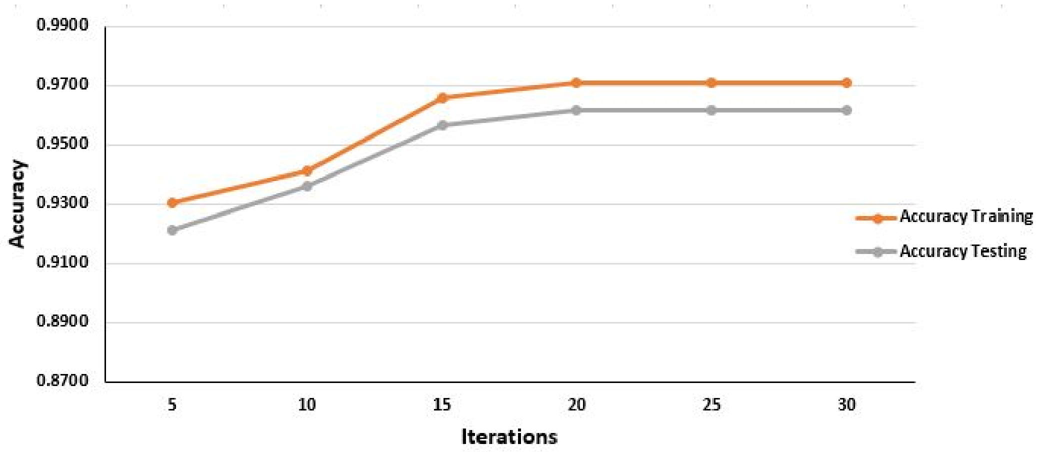

Figure 4.2: Effect of the number of iterations on the accuracy of PSO-Pegasos algorithm for training and testing sets.

According to parameter analysis and paper results, we put number of iterations in PSO = 20 to run the proposed algorithm, therefore, the computational cost is small. Figures 4.3 and 4.4 show the performance measures accuracy, recall, precision and F-measure of Pegasos and PSO-Pegasos algorithms for training and testing sets, respectively. Experimental results in Figures 4.3 and 4.4 show that the accuracy of the proposed algorithm (PSO-Pegasos) for training and testing sets are higher than the accuracy of Pegasos algorithm by about 3.39 and 3.48 respectively. It also shows that the proposed algorithm outperforms Pegasos algorithm in terms of recall, precision and F-measure for training and testing sets due to the existence of particle swarm optimization, which has strong global search capability and high convergence speed to optimal solution.

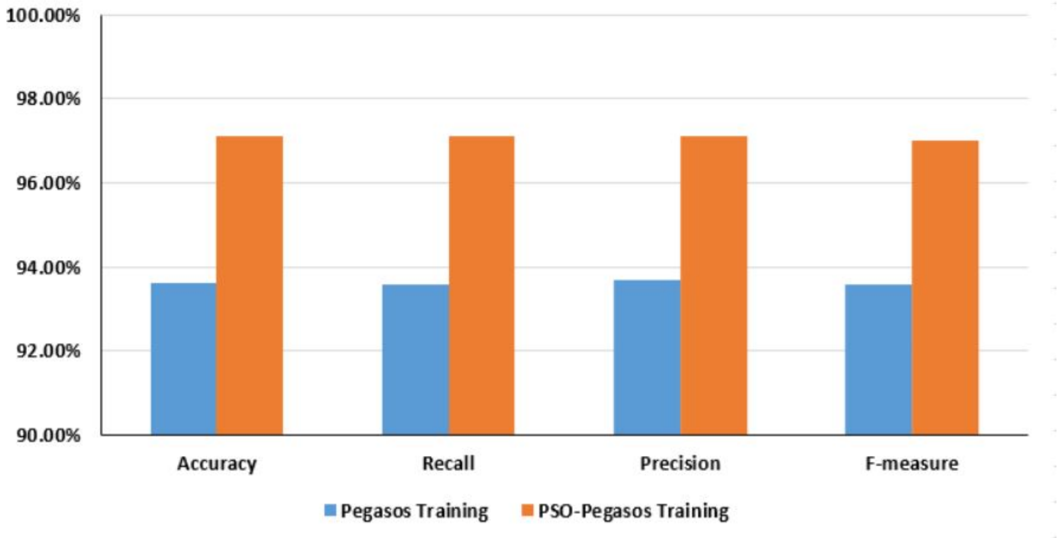

Figure 4.3: Comparison of performance measures of Pegasos and PSO-Pegasos for training set.

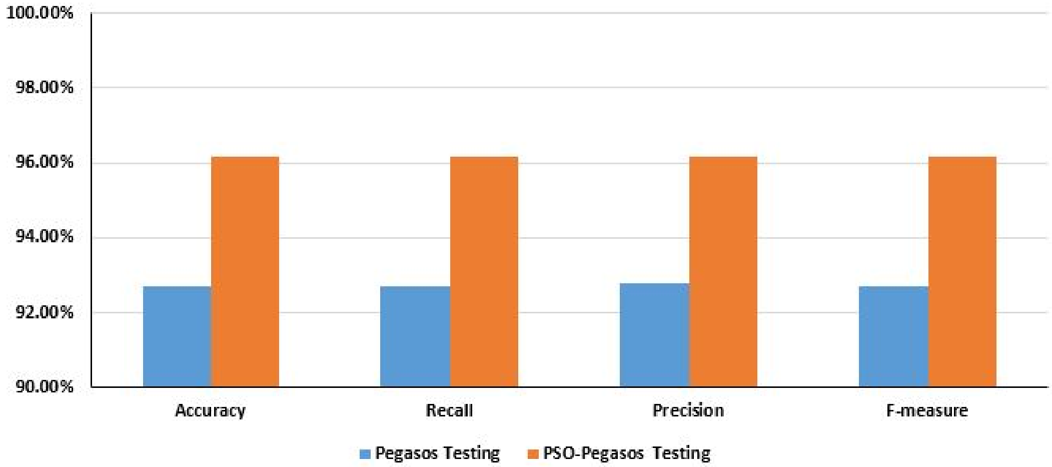

Figure 4.4: Comparison of performance measures of Pegasos and PSO-Pegasos for testing set.

Finally, in order to indicate that the improvement obtained by the proposed algorithm (PSO-Pegasos) clearer, its accuracy compared with earlier used algorithms implemented on the same dataset is presented below.

Experimental results in Table 4.3 show that the results of the proposed algorithm outperforms the results of other published classifier called SVM-based spam detector [11]. The proposed PSO-Pegasos presented improvement of 2.13% over SVM-based spam detector model, which is the best among the other earlier published classifiers for spam email

Table 4.3: Comparison of accuracy of the proposed algorithm and other published classifiers on spambase dataset

classifiers	Classification Accuracy
GA-Naive Bayes [6]	77%
ACO-Naive Baye [6]	84 %
PSO-LM [9]	90.5 %
NSA [10]	68.86 %
PSO [10]	81.32 %
NSA-PSO [10]	91.22 %
HC-RBFPSO [1]	91.4 %
SVM-based spam detector [11]	94.06%
PSO-Pegasos (proposed)	96.19 %

detection. It also presented an accuracy improvement of 4.79 % over a hybrid approach (HC-RBFPSO), that combines radial basis function neural network (RBFNN) and particle swarm optimization (PSO) algorithm [1]. The proposed algorithm also presented improvement of 4.97% over a hybridized negative selection algorithm and particle swarm optimization (NSA-PSO) [10], yet the proposed algorithm in this paper outperformed all the three algorithms NSA-PSO, NSA and PSO including the hybrid schemes. It also presented an accuracy improvement of 5.69 % over learning method for process neural networks based on particle swarm optimization (PSO-LM) [9] and an accuracy improvement of 12.19 % over hybrid ant colony optimization and naive bayes (ACO- Naive Bayes), while presenting an accuracy improvement of 19.19% over hybrid genetic algorithm and naive bayes (GA- Naive Bayes) [6].

5. CONCLUSION

Primal estimated sub-gradient solver for SVM (Pegasos) algorithm was utilized to solve the optimization problem cast by support vector machines (SVM). It is characterized by better convergence bounds and robustly convex optimization objective. In this paper, Hybrid particle swarm optimization (PSO) and Pegasos algorithm, called (PSO-Pegasos) is proposed for spam email detection. PSO is used to identify automatically the best optimal w-parameter for original Pegasos algorithm. The proposed algorithm has been trained and tested using popular and often used spambase dataset, which consists of collection of spam and non-spam emails with 57 features and 1 classification attribute. Excremental results indicated that the proposed PSO-Pegasos algorithm outperformed original Pegasos algorithm and other recently published algorithms tested on the same popular dataset used in this paper. The need for more accurate spam email detection method cannot be overemphasized, the proposed PSO-Pegasos algorithm provides improvement of 2.13% over SVM-based spam detector model, which is the best among the previous reported schemes for spam email detection. The results show that PSO-Pegasos improves the convergence accuracy and it is an effective algorithm, which is a powerful alternative for spam email detection. We can conclude that the aim of this paper has been achieved through training and testing proposed PSO-Pegasos algorithm on spambase dataset. This algorithm has enabled build an improved spam email detection system based on hybridization of particle swarm optimization and Pegasos algorithm.

REFERENCES

1. Awad M, Foqaha M (2016) Email spam classification using hybrid approach of rbf neural network and particle swarm optimization, International Journal of Network Security and Its Applications (IJNSA) Vol.8, No.4, pp. 17-28.
2. Saad O, Hassanien A, Darwish A, Faraj R (2013) A Survey of Machine Learning Techniques for Spam Filtering, IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.1, pp. 103-110.
3. ZhiWei M, Singh M, Zaaba Z (2017) Email spam detection: a method of metaclassifiers stacking, Proceedings of the 6th International Conference on Computing and Informatics, ICOCI, pp. 750-757.757.
4. Sabri A, Mohammads A, Al-Shargabi B, Hamdeh M (2010) Developing New Continuous Learning Approach for Spam Detection using Artificial Neural Network (CLA_ANN), European Journal of Scientific Research, 42(3), pp. 525-535.
5. Zhang Y, Li H, Niranjan M, Rockett P (2008) Applying costsensitive multiobjective genetic programming to feature extraction for spam e-mail filtering. Springer, Berlin, pp. 325-336. doi:10.1007/978-3-540-78671-9_28
6. Renuka D, Visalakshi P, Sankar T, Improving E-Mail Spam Classification using Ant Colony Optimization Algorithm, International Journal of Computer Applications (0975 - 8887)International Conference on Innovations in Computing Techniques (ICICT 2015)
7. Özgür L, Güngör T, Gürgen F (2004) Spam mail detection using artificial neural network and Bayesian filter. pp. 505-510. doi:10.1007/978-3-540-28651-6_74.
8. Temitayo F, Stephen O, Abimbola A (2012) Hybrid GA-SVM for efficient feature selection in E-mail classification, Computer Engineering and Intelligent Systems, Vol 3, No.3, pp. 17-29
9. Liu K, Tan Y, He X (2010) Particle swarm optimization based learning method for process neural networks, In Advances in Neural Networks-ISNN 2010 (pp. 280-287). Springer Berlin Heidelberg.
10. Idris I, Selamat A (2014) Improved email spam detection model with negative selection algorithm and particle swarm optimization, Applied Soft Computing, 22, pp. 11-27.
11. Olatunji S (2017) Improved email spam detection model based on support vector machines, Neural Computing and Applications, 20131(3), pp. 691-699.
12. Kennedy J, Eberhart R (1995)"Particle swarm optimization". In Proceedings International Conference on Neural Networks (ICNN 95) Perth, Australia, pp. 1942-1948.
13. Modares H, Alfi A, Sistani M (2010) Parameter estimation of bilinear systems based on an adaptive particle swarm optimization, Engineering Applications of Artificial Intelligence, 23(7), pp. 1105-1111.
14. Liu Z, Li H, Zhu P (2019) Diversity enhanced particle swarm optimization algorithm and its application in vehicle lightweight design, Journal of Mechanical Science and Technology, 33 (2), pp. 695-709.
15. Cheng S, Lu H, Lei X, Hi Y (2018) A quarter century of particle swarm optimization, Complex and Intelligent Systems, January 2018, Accepted, 22 March 2018
16. Shalev-Shwartz S, Singer Y, Srebro N, (2007) Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, in Proceedings of the 24th International Conference on Machine Learning, pp. 807 -814.
17. Shalev-Shwartz S, Singer Y, Srebro N, Cotter A (2011) Extended version: Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, Mathematical Programming, Series B, 127(1), pp. 3-30, Springer and Mathematical Optimization Society.
18. Lu S, Jin Z (2017) Improved Stochastic gradient descent algorithm for SVM, International Journal of Recent Engineering Science (IJRES), ISSN 2349-7157, Vol 4, pp. 39-42.
19. V. Jumutc, X. Huang, J. A. K. Suykens,(2013) Fixed-size pegasos for hinge and pinball loss SVM, in Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1122-1128, 2013.

20. Hopkins M, Reeber E, Forman G, Suermondt J (1999) SpamBase dataset. Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304. https://archive.ics.uci.edu/ml/datasets/Spambase.
21. El Bakrawy L (2017) Grey Wolf Optimization and Naive Bayes classifier Incorporation for Heart Disease Diagnosis, Australian Journal of Basic and Applied Sciences, 11(7) M, pp. 64-70.
22. Liu P, Moh T (2016) Content Based Spam E-mail Filtering, in Proceedings of the International Conference on Collaboration Technologies and Systems, 978-1-5090-2300-4/16 31.00, IEEE, pp. 2018-2024.