



Feature Selection Empowered by Self-Inertia Weight Adaptive Particle Swarm Optimization for Text Classification

Muhammad Asif, Arfan Ali Nagra, Maaz Bin Ahmad & Khalid Masood

To cite this article: Muhammad Asif, Arfan Ali Nagra, Maaz Bin Ahmad & Khalid Masood (2022) Feature Selection Empowered by Self-Inertia Weight Adaptive Particle Swarm Optimization for Text Classification, Applied Artificial Intelligence, 36:1, 2004345, DOI: [10.1080/08839514.2021.2004345](https://doi.org/10.1080/08839514.2021.2004345)

To link to this article: <https://doi.org/10.1080/08839514.2021.2004345>



© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 04 Dec 2021.



Submit your article to this journal [↗](#)



Article views: 1717



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 4 View citing articles [↗](#)

Feature Selection Empowered by Self-Inertia Weight Adaptive Particle Swarm Optimization for Text Classification

Muhammad Asif^a, Arfan Ali Nagra^a, Maaz Bin Ahmad^b, and Khalid Masood^a

^aDepartment of Computer Science, Lahore Garrison University, Lahore, Pakistan; ^bCollege of Computing and Information Sciences, Paf Kiet, Karachi, Pakistan

ABSTRACT

Text classification (TC) is a crucial practice in case of organizing a vast number of documents. The computational complexity of the TC process is usually high because of the large dimensionality of the feature space. Feature Selection (FS) procedures are used to extract the helpful information from the feature space and results in dimensionality reduction. The development of the FS method that reduces the dimensionality of feature space without compromising the categorization accuracy is desirable. This paper proposes a Self-Inertia Weight Adaptive Particle Swarm Optimization (SIW-APSO) based FS methodology to enhance the performance of text classification systems. SIW-APSO has fast convergence phenomena due to its high search competency and ability to find feature sub-set efficiently. For text classification, the K-nearest neighbors algorithm is used. The experimental analysis shows that the proposed method outperformed the existing state-of-the-art algorithms on the Reuters-21578 data set by achieving 98.60% precision, 96.56% recall, and 97.57% F1 score.

ARTICLE HISTORY

Received 30 September 2020
Revised 31 October 2021
Accepted 4 November 2021

Introduction

In the today's world of big data with large digital documents, Text Classification (TC) has gained tremendous importance, especially for companies to maximize their workflow or even profits. TC is a challenging task that can be used in several applications such as product predictions, movie recommendations, text mining, sentiment analysis, etc. It is one of the popular domains of Natural Language Processing (NLP) that allows a program to classify free-text documents using pre-defined classes. The classes can be categorized either on topic, genre, or sentiments in the text. TC can be conducive practically while considering the vast amount of online text in the form of digital libraries, web-sites and e-mails (Kowsari et al. 2019). The performance of the classifier depends mainly on the feature selection process. If the features are not selected properly, it would affect the prediction accuracy.

CONTACT Arfan Ali Nagra  arfan137nagra@gmail.com  Department of Computer Science, Lahore Garrison University, Lahore, Pakistan

© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The feature space's high dimensionality is the main challenge for TC. The presence of many unique terms and a huge number of the actual terms used are the characteristics of feature space. Thus, reducing the size of the feature space without reducing categorization accuracy is desirable (Ikonomakis, Kotsiantis, and Tampakas 2005).

Feature Selection (FS) is being vastly used in multiple fields to filter out irrelevant and redundant information from the original data (Seal et al. 2015). It reduces the dimensionality of the data set by selecting the appropriate features without compromising the prediction accuracy. If the dimensionality of data is large, the computational complexity of classifiers becomes higher and may have adverse effect on their performance (Zahran and Kanaan 2009). FS algorithms are used in many fields, e.g., categorizing the text, data mining, pattern recognition, digital imaging, and signal processing (Zhao et al. 2010).

To find a sub-set of optimal features is a difficult task as there is no single criterion defined for it (Ge et al. 2016). Generally, the FS procedure consists of sub-set generation, termination criterion, sub-set evaluation, and result validation. The sub-set selection procedure uses a search technique that selects feature sub-set for evaluation purposes depending on a specific search method (Shang et al. 2007). This search method comprises forwarding combination method, backward combination method, forward selection method, and backward elimination method. The selection and evaluation process of the feature sub-set is continuous until the given termination condition is satisfied. After selecting the best feature sub-set, another data set must be validated (Miao and Niu 2016).

The steps involved in selecting feature sub-set can be divided into wrappers, filters, and embedded approaches. The filter's task is to separate FS from the learning algorithm and choose a sub-set that shouldn't depend on any particular learning algorithm (Lee et al. 2019). The evaluation method is used in the wrapper approach to select the feature sub-set. It is based on an exact learning algorithm to be used in the next step. During the evaluation process, the efficiency and sustainability of the sub-sets are checked to find the better one. The comparison of the sub-set with the prior best particle is also part of it. A stopping condition is checked at each iteration to find whether the FS should continue or stop. The wrapper function is considered a better solution generator because it is complex and can break into many features. If the FS and learning algorithm are interleaved, then the FS procedure falls in the domain of the embedded function (Wu et al. 2013).

Particle Swarm Optimization (PSO) is a technique that has a global search ability to select the optimal feature sub-set. There is a problem with this algorithm that it loses its diversity easily, which causes its premature convergence (Karol and Mangat 2013). To overcome the above-mentioned problem, a new algorithm called Self-Inertia Weight Adaptive Particle Swarm Optimization (SIW-APSO) is brought to light. It stabilizes the exploration

capability of the improved inertia weight in PSO. In SIW-APSO, every particle in a loop keeps improving its position and velocity and updates through a developmental process. By keeping all the functionalities of SIW-APSO, it is found better above all the previous versions of PSO (Nagra, Han, and Ling 2019).

In this work, SIW-APSO based feature selection technique is proposed that doesn't require a priori knowledge to work. SIW-APSO improves the premature convergence of the PSO. It has fast rate of convergence and ability to find the feature sub-set efficiently. For text classification, K Nearest Neighbors (KNN) algorithm is used because of its simplicity, quick response and ease of use for multi-class problems. The experimental analysis shows that the proposed method outperformed the existing state-of-the-art algorithms on the Reuters-21578 data set by achieving 98.60% precision, 96.56% recall, and 97.57% F1 score.

The rest of the paper is organized as follows: Section 2 reflects the literature review. Section 3 gives a brief overview of the SIW-APSO algorithm and describes the proposed technique for feature selection. The experimental analysis is performed in section 4. Finally, section 5 concludes the paper.

Literature Review

Several feature selection techniques have been proposed in the literature to for TC systems. Pedersen and Yang (Yang and Pedersen 1997) conducted a study to compare criteria of five FS methods for TC, including mutual information, information gain, document frequency, term strength, and x2-text (CHI). They observed that x2 and information gain are more appropriate for optimizing classification results.

Leveraging Association Rules in FS to classify text is a good feature selection method (Aghbari and Saeed 2021). A hybrid approach is proposed by Lee et al. (Lee et al. 2019) to select multi-label text features. It did improve the learning performance through competition among selective operators. An algorithm is proposed to selectively apply each operator, which rectifies the sub-set of features by its effectiveness and relative efficiency, different from the traditional approaches. Results taken by using multiple text datasets reveal that it outclassed the traditional techniques. A Parallel Global TFIDF Feature Selection Using Hadoop for Big Data Text Classification is a classic feature selection method (Amazal, Ramdani, and Kissi 2021). An application of MOGW optimization for feature selection in text classification is a better method for classification (Asgarnezhad, Monadjemi, and Soltanaghaei 2021).

(Chun-Feng, Kui, and Pei-Ping 2014) represented another algorithm that presented the enhanced version of Artificial Ant Colony (ABC) with PSO, which has poor exploitation. So, a study is proposed to work with PSO to select sub-set in text categorization effectively. For better performance, this

algorithm works in multiple steps. The first step chooses a good point sub-set instead of random selection, which improves the convergence speed. Secondly, the ability of exploitation is enhanced by utilizing PSO to search for a new sub-set. Finally, a disordered search process takes place for the solution of recent iteration, which increases the effectiveness of searchability. The proposed algorithm is compared with different other algorithms; results show its better performance than others.

Khem et al. (Khem and Tong 2008) observed three techniques, i.e., centroid, LDA/GSVD, and orthogonal-centroid. These techniques are designed for a dimensional reduction in TC by minimizing the dimensions of clustered data. Similarly, an amazing review of FS for TC is presented by Forman (Forman 2007), and a case study is introduced for the selection of text features.

Zhang et al. introduced a PSO-based multi-objective and multi-label feature selection algorithm. The algorithm specifies the probability-based encoding, which transforms the nature of the feature selection problem into a continuous feature selection issue suitable for PSO. The specified algorithm is evaluated and contrasted with other methods, like RF-BR and MI-PPT. The results reveal that the algorithm can search for the best subset (Zhang et al. 2017).

Original PSO is expected to give an optimal feature sub-set due to its simplicity in searching in a one-dimensional search space. Multi Swarm (MSPSO) is very effective because it outclasses genetic algorithm (GA) and other rival algorithms, e.g., standard PSO, grid search, etc. Several variants of PSO are presented in the literature using filter and wrapper techniques, all aiming to make feature selection more efficient. After sufficient investigations, it is observed that PSO and its variants are relatively more effective during the selection process of a sub-set containing optimal features (Vashishtha 2016).

Radial Basis Function (RBF) network is used by Bilal et al. (Zahran and Kanaan 2009) as a text classifier. The proposed technique is compared with the efficiency of document frequency, TF-IDF, and Chi-square statistic algorithms. Results taken from the Arabic dataset determined the dominance of the proposed algorithm. The fitness function, Inertia parameters (w), and position-updating strategy are critical performance parameters of PSO. There are three major contributions of this work: Initially, evolutionary algorithms are overviewed using explicit or implicit memory, which is applied to dynamic fitness functions. Secondly, it suggests a new benchmark by observing the previous test problems to fill the gap between simple and complex real-world applications. The proposed benchmark is based on the derivation of Branke (Branke 1999), which is not limited to memory-based approaches. It is a fresh way to discover the advantages of a memory-based system while reducing its side effects.

James (Eberhart and Kennedy 1995) described the relationships between artificial intelligence and particle swarm optimization and evolving computation. The testing benchmark of both paradigms is discussed, and applications are proposed that comprise training neural networks and learning the tasks for robots. Three kinds were tested: the first one is the “GBEST” model, where each agent knows the best evaluation of the group. The rest were the two variations of the “LBEST” type, both of those with the neighborhood of six and two, respectively. The first one comes up with the actual version of GBEST that performs exceptionally well in quick convergence. The other version of LBSET (with the neighborhood of two) is the most impervious to local minima.

Exhaustive searching is the simplest way to discover the optimal subset of features through the evaluation of all sub-sets. But this is impractical to some extent, even for a moderate-sized feature set. To avoid this complexity, the FS method normally involves random search strategies. Therefore, the final feature sub-sets optimality is often minimized. Among most of the methods suggested for the FS, genetic algorithm, ant colony optimization and PSO have been emphasized more by the researchers. By collecting knowledge from previous steps, these methods try to get better solutions. GAs is a method for optimization based on natural selection. In search space, they implement methods found in natural genetics. In data mining, GAs have been used at a large scale as a tool for FS because of their advantages. PSO was first presented by Kennelly and Eberhart as a model for social learning and influence. In the Swarm-based method, particles follow a very basic mechanism, i.e., contend with the success of nearby particles. The multidimensional search space’s optimal regions are found through the emerging cluster behavior.

Aghdam and Heidari presented a modified PSO-based FS technique. In this technique, a document is visualized as a set of phrases or words. The feature vector’s all positions tally the given terms of the document. The evaluation is done using the bag of words model on text features. Abdelhalim et al. proposed a novel hybrid technique by using PSO and Nelder–Mead (NM) simplex search algorithm. It was intended to give answers to nonlinear unconstrained optimization problems. This method improved the positions of the particles and the velocities by using the NM scheme inside the PSO method. The evaluation of the suggested technique was done by using more than 20 optimization test functions having variable dimensions. Further, the detailed comparison with some other techniques highlights that the suggested technique is not only reliable but also a competitive one.

In nutshell, most of the existing feature extraction techniques for text classification demand a prior knowledge that requires additional computational resources. Moreover, some of the techniques also face the issue of premature convergence. To address aforementioned issues, this work presented SIW-APSO based feature selection approach.

Proposed SIW-APSO Based Feature Selection Technique

Overview of SIW-APSO

The SIW-APSO is a modified version of the PSO technique. It improves the premature convergence of PSO. In this method, a random swarm of particles is initialized, and each particle represents a position in the search space, and a fitness function is determined from the pre-evaluated function, which is very much helpful to gain the required optimal solution. The particles move around about a dimensional solution space in the swarm, velocity and position of a particle are updated with the help of equation 1 and equation 2, respectively.

$$S_{id}^{t+1} = W_i^{t+1} S_{id}^t + k_1 r_1 \left(\frac{(\beta f(Z_i^{t+1})) - (\beta f(Z_i^t))}{(\beta f(Z_i^{t+1}))} \right) + k_2 r_2 (p_g^t - Z_i^t) \quad (1)$$

$$Z_{id}^{t+1} = S_{id}^{t+1} + Z_{id}^t \quad (2)$$

where Z_i^d is d th position of particle i , W_i^{t+1} is the inertia weight of i th at the iteration $(t + 1)$, k_1 & k_2 are the constants, r_1 & r_2 the random functions, and $p_b^{d''}$ is the d th best position particle i , and is the d th global best position founded by the whole swarm. According to equation 3, the iterative behavior of inertia weight is very much helpful to find accurate fitness.

$$W_i^{t+1} = 0.9ift = 0 \quad (3)$$

$$W_i^{t+1} = f(Z_i^t) - f(Z_i^{t-1})ift > 0 \quad (4)$$

where W_i^{t+1} and $f(Z_i^t)$ represent the inertia weight and best fitness value at the t^{th} and $t + 1^{\text{th}}$ iteration, respectively.

As in equation (1), $\frac{(\beta f(Z_i^{t+1})) - (\beta f(Z_i^t))}{(\beta f(Z_i^{t+1}))}$ represent the enhancement in the best fitness function that is dependent on proposed inertia weight. Equation (4) shows the $f(Z_i^t)$ and (Z_i^{t-1}) fitness function of the swarms at the t^{th} and $t - 1^{\text{th}}$ repetitions, respectively. If $W_i^{t+1} = W_i^t$, the one thing is clear that the particle unable to find an optimal solution because $f(Z_i^{t-1}) = f(Z_i^t)$, and nothing improve in inertia weight. In the case of, as we can use in the above equation (4), then the variance in inertia weight is optimistic, and the fitness will be improved. The fitness function progresses if the finest position is right close to the other particle in the swarm. W_i^t will grow to W_i^{t+1} at $t - 1^{\text{th}}$ loop, which is obliging for exploration and exploitation. When $W_i^{t+1} = 0.9$, as described in equation 3, the inertia weight is persistent and acts similar to an ordinary PSO.

In SIW-APSO, the particle begins the optimization with $w = 0.9$, as this value is shown in equation (3). Inertia weight updated using equation (4) which shows the global best fitness. For optimization functions, there are great

vibrations in the value of inertia weight in the very first iterations, which are beneficial for the particle to sustain its diversity. The oscillations decreased toward the end of iterations, which resulted in fine changes in the desired solution. The SIW-APSO can achieve in good manners to enhance the accuracy of PSO.

Through the research, we find the on some stages, inertia weight is equal to zero and does not grow in several uninterrupted iterations, which shows that swarm stuck cause the stall in a local optimum. This condition may be directed toward premature convergence because the current position of the particle is lower than a predetermined threshold value. For this shortfall, a linear function is used to map the possible range of the inertia weight values:

$$W(t) = W_l - (W_l - W_h)t/t_h \quad (5)$$

where w_i is the initial value of inertia weight, w_h is the final value of w_i , (t) is the current iteration of the algorithm, t_h is maximum number of iteration, and the range of inertia weight is [0,1,2]. According to equation (4), the inertia weight for every swarm at each iteration is changed independently, using the enhancement in its personal best fitness. There is an alert for change of fitness; at any iteration, when particles' personal best fitness improves, then the particle changes its current direction; else its inertia weight is set to zero, and particle starts its search locally and improvement in inertia which is global exploration.

The inertia weight of each particle is updated individually, so it is also possible for all particles in the swarm to have different inertia weights; finally, some particles can search locally and some globally. So far, balancing between global and local research according to the mentioned method improves the diversity and is very useful for exploration. The algorithmic flow of SIW-APSO algorithm is given in [Figure 1](#).

SIW-APSO for Feature Selection

This approach optimizes the problem in a continuous, multidimensional search space. SIW-APSO starts with a group of arbitrary particles. The behavior of each particle is adjusted according to its velocity. These particles have a propensity of moving toward the better search spaces. The SIW-APSO algorithm is described mathematically in the above equations (1) and (2).

SIW-APSO is deliberated for searching multi-dimensional continuous search spaces. In this work, the feature selection problem has been adopted in which each feature sub-set considers as a point in feature space. The initial swarm scattered arbitrarily over the search space. Every particle takes one location, and the aim of particles is to transfer the finest position. Particles change their position by communicating with each other, and they search for

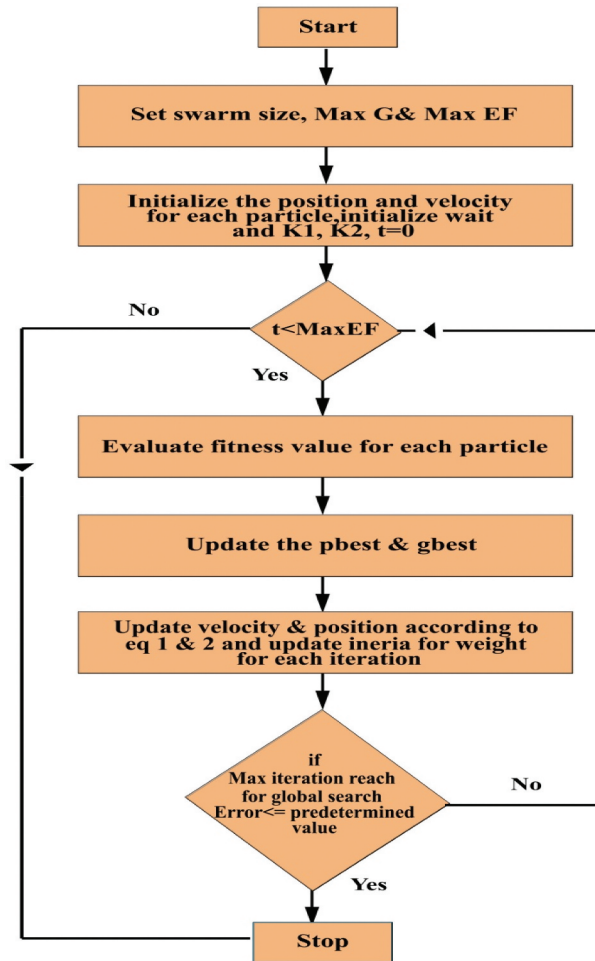


Figure 1. Work flow of a SIW-APSO algorithm.

the local best and global best positions. Lastly, they reach their decent possible optimal position; meanwhile, they have exploration aptitude that performs the FS and optimal subset.

SIW-APSO needs to extend in direction to treat with feature selection. The particle's position is considered as a binary bit vector, where the bit value 1 is considered as a selected feature, and another bit 0 is represents the non-selected feature. Each position is a feature subset; hence the SIW-APSO is effectively used to optimize the problem. The algorithmic flow of the proposed feature selection algorithm is shown in [Figure 2](#).

The first step after we input the text documents in the system is the preprocessing step which is very important in text classifications; in this step, the stop words are removed, stemming is applied to the remaining text in the document, this helps reduce the feature space of the problem.

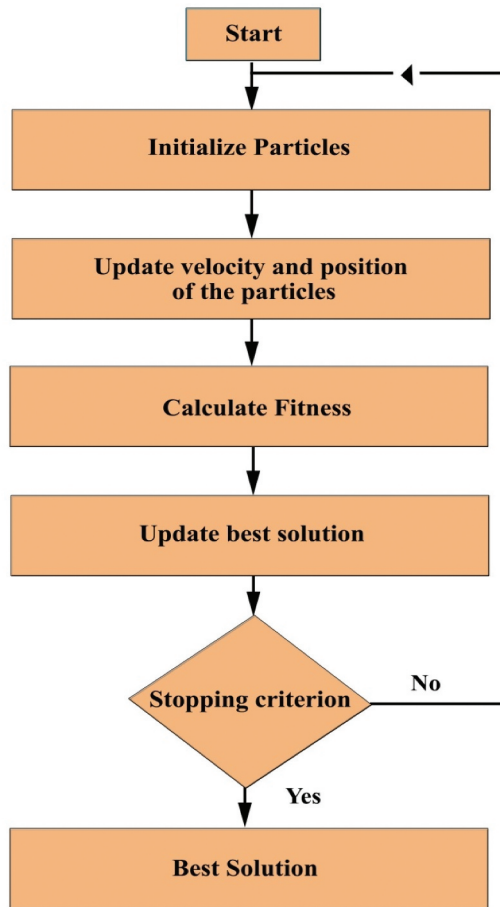


Figure 2. Proposed SIW-APSO feature selection algorithm.

Initializing Particles

In this method, every particle is represented by a binary vector. The binary vector has an equal length to the actual features. The binary vector value is set to 1 when the required feature is selected and when the required feature is not selected, its value is set to 0. In the proposed method, the desired features are a subset of extracted feature set. After that, a velocity vector was produced for each particle. Thus, the velocity vector has the same as the particle vector length. Random value in the range of $[0-1]$ is set for each cell of the velocity vector.

Updating Velocity

Each particle has a velocity which is shown such as a positive number, the velocity of the particle is bounded to show that how many features have been changed or same as an optimal point. The velocity of particles moving toward the best position. In equation 1, velocity is updated in this novel variant of PSO.

Updating Position

Update the position of the particles with the help of updating velocity, so the position of the particle will be updated by the new velocity best. If we have a new velocity is V and V , bits of particles are arbitrarily altered different from gbest. The particle moved toward the global best while exploring the search area instead same as gbest. In SIW-APSO, each particle changes its position according to its velocity, as depicted in equations 1 and 2.

Fitness Function

The fitness function has been explained through equation 6:

$$\text{Fitness}(X_i) = \delta \cdot \gamma(F_i(t)) + \psi \cdot (n - |F_i(t)|) \quad (6)$$

In this equation, $F_i(t)$ is the feature subset at iteration i , originating by swarm size i with the iteration length shown as $|F_i(t)|$. Fitness is calculated in order to together measure the accuracy of a classifier, $\gamma(F_i(t))$, and the feature subset area. and ψ are the two parameters that have been managed the relative weight of the classifier performance and the feature subset area, $\delta \in [0, 1]$ and $\psi = 1 - \delta$. According to this formula, both classifier performance and feature subset area have many effects on FS. The proposed method considers the classifier performance combined with sub-set length, so we set them to $\delta = 0.8$, $\psi = 0.2$. [Figure 2](#) illustrates the complete steps of the proposed algorithm for feature selection in text categorization.

Experimental Results

In this work, MATLAB is used to implement the proposed algorithm. To evaluate the performance of the proposed feature selection technique, several experiments are conducted on an Intel Core i7 machine with a window 10.0 operating system, 3.6 GHz CPU, and 8 GB RAM. For experimental analysis, the top ten most frequently used classes of the Reuters –21578 documents dataset (Lewis 1997) are considered. [Table 1](#) illustrates the classes and number of documents that belong to each class. The dataset is divided into two sets, one for training and the other for testing the classifier. The training and testing set consist of 5785 and 2299 documents, respectively. The number of documents samples taken from each class for training and testing is unbalanced.

To demonstrate the efficacy of the proposed feature selection approach, KNN classifier is used [24]. For the classifier, it is not possible to directly interpret the text documents. To address this problem, there is a need to uniformly apply indexing procedures on documents that transform their content into compact representation. In compact representation, a text T_j is generally indicated by a feature vector of term weights. Each location in the

Table 1. Classes details.

Sr. No	Classes	Training Samples	Testing Sample	Total Samples
1	Grain	72	32	104
2	Ship	122	42	164
3	Wheat	153	51	204
4	Interest	165	74	239
5	Corn	170	53	223
6	Crude	288	126	414
7	Trade	297	99	396
8	Money-fx	313	106	419
9	Acquisition	1484	664	2148
10	Earn	2721	1052	3773

feature vector is equivalent to the specified phrase or word. This indication is known as the bag of words model. In this work, the normalized *TF-IDF* function given in the equation is used to calculate the weights.

$$P_{kj} = \frac{TF - IDF(f_k, T_j)}{\sqrt{\sum_{m=1}^L (TF - IDF(f_s, T_j))}} \quad (7)$$

where f_k indicates the features set that come at least once in the training documents and $0 \leq P_{kj} \leq 1$ denotes the contribution of FK to the semantics of document T_j .

$$TF - IDF(f_k, T_j) = N \cdot \log\left(\frac{L}{N_d}\right) \quad (8)$$

In equation 8, the number of occurrence f_k in T_j are indicated by N . T_s indicates the training set, and L denotes its length. N_d represents the documents count in T_s in which f_k comes.

To evaluate the performance of the classifier following parameters are used:

$$Precision(\rho_i) = \frac{\tau\rho_i}{\tau\rho_i + \rho_i} \quad (9)$$

$$Recall(R_i) = \frac{\tau\rho_i}{\tau\rho_i + \mathcal{N}^i} \quad (10)$$

$$F1 - Score(F_i) = \frac{2 * R_i * \rho_i}{(\rho_i + R_i)} \quad (11)$$

where $\tau\rho$ (True Positive) and ρ (False Positive) denote correctly and incorrectly classified documents underclass i , respectively. The number incorrectly classified document of class i to other classes are represented by \mathcal{N} (False Negative). These parameters or probabilities are computed with the help of a contingency table for each class C_i on the specified dataset.

In the case of multiple classes, micro average and macro average can be computed for the above-mentioned parameters. In the case of macro average, all classes are treated equally regardless of the number of documents that belong to that class. On the other hand, in the case of micro average, all the documents are weighted equally, therefore favoring the performance of common classes. Equations 12–15 indicate macro and micro averaging on recall and precision. The global contingency table is shown in [Table 2](#), which is determined through the addition of class-specific contingency tables.

$$\rho^\mu = \frac{\sum_{i=1}^{|c|} \tau \rho_i}{\sum_{i=1}^{|c|} (\tau \rho_i + \rho_i)} \quad (12)$$

$$R^\mu = \frac{\sum_{i=1}^{|c|} \tau \rho_i}{\sum_{i=1}^{|c|} (\tau \rho_i + \rho_i)} \quad (13)$$

$$\rho^M = \frac{\sum_{i=1}^{|c|} \rho_i}{|C|} \quad (14)$$

$$R^M = \frac{\sum_{i=1}^{|c|} R_i}{|C|} \quad (15)$$

where μ and M represent the macro and micro average, respectively.

The comparative analysis of the proposed technique is made with four previous studies for feature selection which are based on PSO, CHI, Information Gain (IG), and Genetic Algorithm (GA). To perform the experimental analysis, the population size is set to 50; the maximum numbers of iterations are adjusted to 100, the value of C1 and C2 is set to 1, and the w is taken in the range of [0.4, 1.4]. [Table 3](#) shows that the proposed technique outperforms the existing techniques in terms of precision, recall, and F1-score. To graphically demonstrate the progress of the proposed feature selection techniques to determine the optimal solution, a graph is drawn between the percent feature and F1 measure, which shows the improvement process of the best particle with respect to the increase in the number of features.

[Figure 3](#) and [Figure 4](#) illustrate the macro and micro averaged F1 score against a number of selected features for all feature selection techniques. These graphs show that proposed techniques supersede the existing techniques as the percentage of selected features exceeds 12%.

[Table 4](#) gives the micro and macro F1 score-based comparison of the proposed technique with existing techniques. It illustrates the best performance is achieved by the SIW-APSO.



Table 2. The global contingency table.

Class set $C = \{c_1, \dots, c_l\}$	Judgments		Expert judgment	
	Yes	No	Yes	No
Classifier	Yes	No	$\tau p = \sum_{i=1}^{ c } \tau p_i$	$\rho = \sum_{i=1}^{ c } \rho_i$
judgment	No	No	$= \sum_{i=1}^{ c } N_i$	$\tau = \sum_{i=1}^{ c } N_i$

Table 3. Performance comparison of the proposed technique with existing techniques.

Class	IG			CHI			GA			PSO			SIW-APSO		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Acquisition	91.75	71.99	80.68	89.56	76.20	82.34	92.45	81.17	86.62	91.53	90.56	90.78	93.38	92.72	93.05
Corn	87.80	67.92	76.60	87.80	67.92	76.60	90.70	73.58	81.25	91.63	74.21	82.01	93.67	78.34	85.32
Crude	84.90	71.43	77.58	83.02	69.84	74.96	78.32	88.89	83.27	81.69	88.89	78.55	84.22	90.82	87.40
Earn	84.38	94.49	89.15	84.98	94.68	89.57	90.20	96.20	93.76	97.56	94.33	95.92	98.60	96.56	97.57
Grain	63.64	65.63	64.62	60.61	62.50	61.55	69.70	71.88	70.77	67.26	76.36	71.52	72.56	80.98	76.54
Interest	45.76	36.49	40.60	54.00	36.49	43.55	60.00	36.49	45.38	50.68	52.64	51.64	55.02	56.33	55.67
Money-fx	51.72	56.60	54.05	61.22	56.60	59.13	67.89	69.81	70.94	68.49	70.81	69.63	72.45	73.78	73.11
Ship	33.82	54.74	44.83	37.50	57.14	45.28	57.14	66.67	61.54	60.35	75.36	67.03	63.75	79.46	70.74
Trade	68.70	90.91	78.26	73.98	91.92	81.98	72.36	89.90	80.18	78.63	91.88	84.74	81.56	93.59	87.16
Wheat	91.30	82.35	86.60	89.36	82.35	85.71	87.76	84.31	86.00	86.79	85.36	84.59	89.37	88.64	89.00
Average	70.38	69.25	69.30	72.20	69.57	70.07	76.65	75.89	75.97	77.46	80.04	77.64	80.46	83.12	81.77

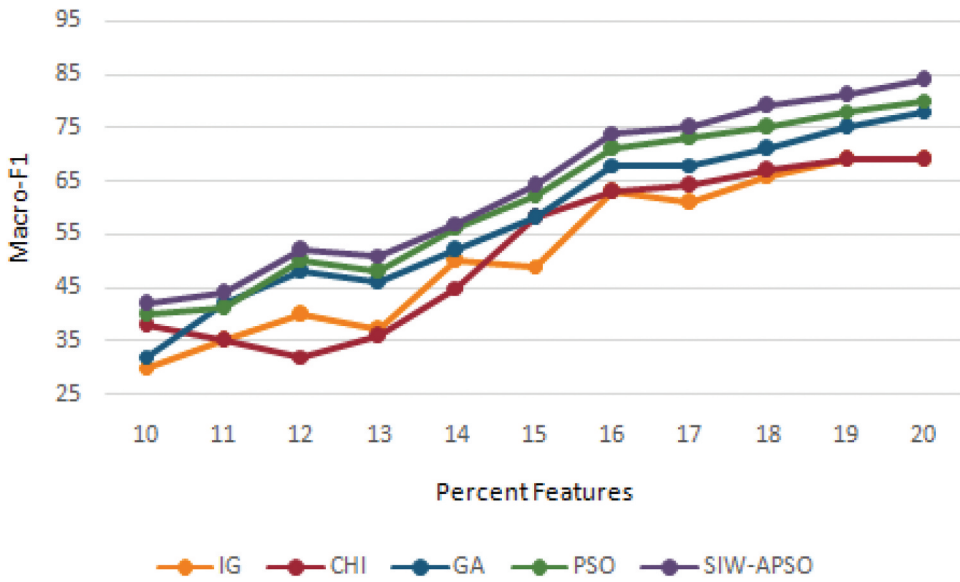


Figure 3. Performance comparison based on Macro-F1 score.

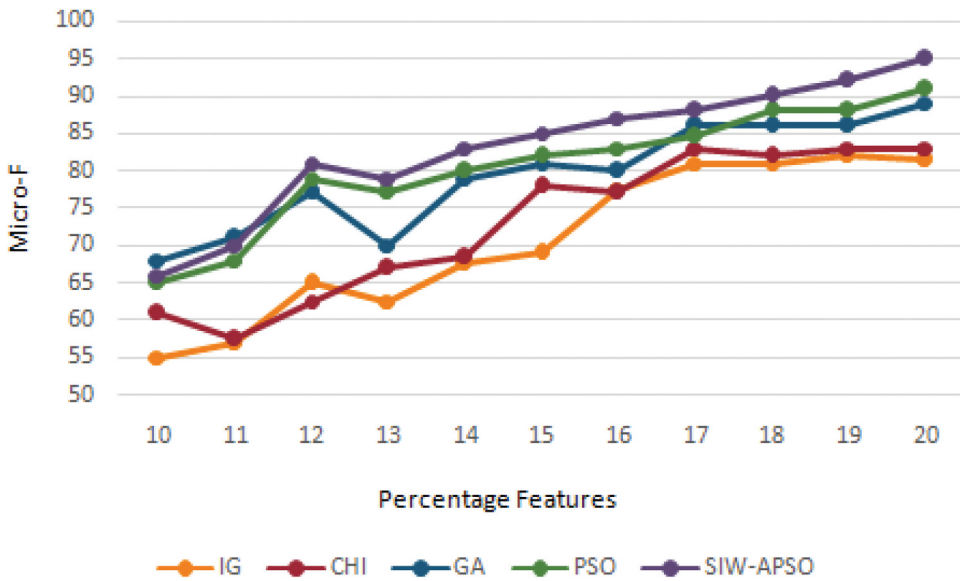


Figure 4. Performance comparison based on Micro-F1 score.

Table 4. Micro F1 and Macro F1 scores of the proposed and existing techniques.

Techniques	μ F1 score	M F1 score
IG	80.95	69.81
CHI	82.21	70.86
GA	86.39	76.27
PSO	89.57	78.86
SIW-APSO	94.43	84.23

In comparison with the existing techniques, SIW-APSO quickly determines the optimal solution. Generally, it determines the optimal solution within tens of iterations. The exhaustive searching is not possible to apply on Reuters-21578 data set to determine the optimal feature subset because of the existence of billions of candidate subsets. On the other hand, in the case of SIW-APSO, the optimum solution is found at the 100th iteration.

Conclusion

This paper proposed the FS technique based on SIW-APSO. First of all, the solution space is searched by it. After that, the evolutionary process is used to iteratively update the speed and position of each particle. The selected feature subset's length and classifier performance are used a heuristic data. To show the efficacy of the proposed technique, the comparative analysis is performed with four existing techniques based on PSO, CHI, Information Gain (IG), and Genetic Algorithm (GA). The experimental results show that the proposed technique outperforms all of its competitors on the Reuters-21578 data set by achieving 98.60% precision, 96.56% recall, and 97.57% F1 score. In future, this work can be applied to other classification problems.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- Aghbari, Z. A., and M. M. Saeed. 2021. Leveraging association rules in feature selection to classify text. In *Computer networks and inventive communication technologies* Smys, S., 715–22. Singapore: Springer.
- Amazal, H., M. Ramdani, and M. Kissi. 2021. A parallel global tfidf feature selection using hadoop for big data text classification. In *Advances on smart and soft computing* Saeed, Faisal, 107–17. Singapore: Springer.
- Asgarnezhad, R. S., A. Monadjemi, and M. Soltanaghaei. 2021. An application of MOGW optimization for feature selection in text classification. *The Journal of Supercomputing* 77 (6):5806–39. doi:10.1007/s11227-020-03490-w.
- Branke, J. 1999. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* USA, IEEE, vol. 3, pp. 1875–82.
- Chun-Feng, W., L. Kui, and S. Pei-Ping. 2014. Hybrid artificial bee colony algorithm and particle swarm search for global optimization. In *Mathematical problems in engineering*, vol. 2014 (Hindawi)8 pages .
- Eberhart, R., and J. Kennedy. 1995. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Ieee, pp. 39–43.

- Forman, G. (2007). Chapter: Feature selection for text classification book: Computational methods of feature selection Chapman and Hall/CRC press, 2007.
- Ge, R., M. Zhou, Y. Luo, Q. Meng, G. Mai, M. Dongli, G. Wang, and F. Zhou. 2016. McTwo: A two-step feature selection algorithm based on maximal information coefficient. *BMC Bioinformatics* 17 (1):1–14. doi:10.1186/s12859-016-0990-0.
- Ikonomakis, M., S. Kotsiantis, and V. Tampakas. 2005. Text classification using machine learning techniques. *WSEAS Transactions on Computers* 4 (8):966–74.
- Il-Seok, O., and J.-S. Lee. 2009. Ant colony optimization with null heuristic factor for feature selection. In *TENCON 2009-2009 IEEE Region 10 Conference*, IEEE, pp. 1–6.
- Karol, S., and V. Mangat. 2013. Evaluation of text document clustering approach based on particle swarm optimization. *Open Computer Science* 3 (2):69–90. doi:10.2478/s13537-013-0104-2.
- Khew, S. T., and Y. W. Tong. 2008. Template-assembled triple-helical peptide molecules: Mimicry of collagen by molecular architecture and integrin-specific cell adhesion. *Biochemistry* 47 (2):585–96. doi:10.1021/bi702018v.
- Kowsari, K., K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. 2019. Text classification algorithms: A survey. *Information* 10 (4):150. doi:10.3390/info10040150.
- Lee, J., J. Park, H.-C. Kim, and D.-W. Kim. 2019. Competitive particle swarm optimization for multi-category text feature selection. *Entropy* 21 (6):602. doi:10.3390/e21060602.
- Lewis, D. (1997). Reuters-21578 Text Categorization Collection Distribution 1.0. <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>
- Miao, J., and L. Niu. 2016. A survey on feature selection. *Procedia Computer Science* 91 (Supplement C):919–26. doi:10.1016/j.procs.2016.07.111.
- Nagra, A. A., F. Han, and Q. H. Ling. 2019. An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search. *Engineering Optimization* 51 (7):1115–32. doi:10.1080/0305215X.2018.1525709.
- Seal, A., S. Ganguly, D. Bhattacharjee, M. Nasipuri, and C. Gonzalo-Martin. 2015. Feature selection using particle swarm optimization for thermal face recognition. In *Applied computation and security Systems* Saeed, Khalid, 25–35. New Delhi: Springer.
- Shang, W., H. Huang, H. Zhu, Y. Lin, Q. Youli, and Z. Wang. 2007. A novel feature selection algorithm for text categorization. *Expert Systems with Applications* 33 (1):1–5. doi:10.1016/j.eswa.2006.04.001.
- Vashishtha, J. 2016. Particle swarm optimization based feature selection. *International Journal of Computer Application* 146 (6):11–17.
- Wu, B., M. Zhou, X. Shen, Y. Gao, R. Silvera, and G. Yiu. 2013. Simple profile rectifications go a long way. In *European Conference on Object-Oriented Programming*, Heidelberg, Springer, Berlin, pp. 654–78.
- Yang, Y., and J. O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Icml* 97 (412–420):35.
- Zahran, B. M., and G. Kanaan. (2009). Text feature selection using particle swarm optimization algorithm 1.
- Zhang, Y., D.-W. Gong, X.-Y. Sun, Y.-N. Guo, Z. Wang, K. Tamada, T. Takumi, R. Hashimoto, H. Otani, and G. J. Pazour. 2017. A PSO-based multi-objective multi-label feature selection method in classification. *Scientific Reports* 7 (1):1–12. doi:10.1038/s41598-016-0028-x.
- Zhao, Z., F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu. (2010). Advancing feature selection research. In *ASU feature selection repository*, Li, Jundong, 1–28.