



A novel hybrid multi-thread metaheuristic approach for fake news detection in social media

Gungor Yildirim¹

Accepted: 5 July 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In fake news detection, intelligent optimization seems to be a more effective and explainable solution methodology than the black-box methods that have been extensively used in the literature. This study takes the optimization-based method one step further and proposes a novel, multi-thread hybrid metaheuristic approach for fake news detection in social media. The most innovative feature of the proposed method is that it uses a supervisor thread mechanism, which simultaneously monitors and improves the performance and search patterns of metaheuristic algorithms running parallel. With the supervisor thread mechanism, it is possible to analyse different key attribute combinations in the search space. In addition, this study develops a software framework that allows this model to be implemented easily. It tests the performance of the proposed model on three different data sets, respectively containing news about Covid-19, the Syrian War, and daily politics. The proposed method is evaluated in comparison to the results of fifteen different well-known deep models and classification algorithms. Experimental results prove the success of the proposed model and that it can produce competitive results.

Keywords Fake news detection · Metaheuristic · Multi-threading · Optimization

1 Introduction

The evolution of the print media to an online broadcasting model, together with the transformation of social networks into news producing-consuming platforms is causing an enormous flow of both news and data, regarding volume, velocity, and variety [36]. Since production and distribution is rather easy, there is nowadays an acceleration of the circulation of fake or fabricated content, as well as real content. This rapid change and transformation in communication technologies leads to people to be vulnerable to the news and data flow they are exposed to. Every day, and almost every second, huge amounts of information and news are presented to millions of users on different platforms, while most of them are unverified. Consequently, people often spread the news they encounter online without seeking any verification, causing fake/false/fabricated information and content to spread at a high speed [41, 42]. This spread has major negative effects on individuals and societies, which include fear, hopelessness,

anger, and prejudice. From a holistic point of view, in addition to its individual and social effects, many other negative effects are observed; from reputational and commercial losses of social media companies, to social polarization, to triggering regional and international crises, to manipulation of political and commercial activities, or to the creation of insecurity in the society. Nowadays, fake news has the ability to control and to shape the scientific, social, and religious realities, among other beliefs and relations. Recently, fake news started its transformation into an asymmetrical attack tool by making use of social engineering elements. As a result of the search for a solution to this problem, a new interdisciplinary research area has emerged in the last 5 years, fake news detection. It has aroused curiosity in many different disciplines, but more specifically computer science.

The content of news usually consists of text, audio, visual, and video components. Considering the dissemination speed, size, diversity, and heterogeneity of data in the online environment, it is a very difficult task to develop an inclusive and systematic tool that can detect the truth of the news. In this context, the analysis of text, which is the most dominant type of content among the components that make up the news, is of great importance. The primary point of reference for the analysis of these text contents is the Natural Language Processing (NLP) discipline. The detection of intentionally misleading

✉ Gungor Yildirim
gungor.yildirim@firat.edu.tr

¹ Department of Computer Engineering, Firat University, Elazig, Turkey

news is based on the analysis of several examples of both fraudulent and truthful previously reviewed news. The spread of fake news in different channels, especially online platforms, was not yet completely stopped or reduced to some extent. This happens because there is no system that can completely check for fake news with little or no human intervention.

There are a number of different fake news detection models, which can be categorized as illustrated in Fig. 1. Expert-fact-checkers are a small group of professionals in a variety of disciplines, who have the ability to confirm the accuracy of certain news, deciding afterwards whether that information is fake or genuine. The advantage of expert verification techniques is that they are easy to manage and highly accurate [47]. However, this expert-fact-checker technique is time-consuming, especially in cases where a large amount of information is given for their verification, due to their small number and also the need for manual annotators. As an alternative to expert-fact-checker, the crowdsourcing methodology relies on the collective approval of individuals or groups [31]. The crowd is made up of ordinary people from diverse backgrounds that has little knowledge of some of the news sites; and as a result, news sites they are not familiar with are flagged as untrusted. The major strength of crowdsourcing methods is that they are based on pluralism, since they use individuals with a diverse knowledge. On the other hand, the number of users actually affected by the event needs to be verified for reliability prior to the classification.

Previous studies show that machine learning algorithms may be capable of detecting fake news, having into account the high number of cases needed to be trained for these models to properly work [28]. These artificial intelligence (AI)-based systems seem efficient methods to automatically verify reality and/or detect fake news [16, 34]. The previously cited works have used AI and human analysis together, which was able to correctly block and tag social media accounts, as well as content, in some important situations and events that was flagged as fake. They also collaborate with fact-checking

organizations that carry out intensive manual processes. In recent years, many researchers have carried out important studies to develop automated solutions to detect fake news, applying different methodologies [14, 23]. At this point, it is important to state that although important steps have been taken to solve the problem, it cannot be said that satisfaction levels have yet been reached. The increase in the number of news verification organizations and the number of people working around this topic [30] can be considered as an indication that suitable (or optimal) solutions have not yet been found. Black-box nature, problems with computational efficiency and low interpretability seem to be the major disadvantages of these methods. Also, these models need a high degree of optimization to get the suitable values of the entire parameters.

The recommendation systems aim to validate news content that is considered original, and afterwards recommend those news articles that are ready for consumption [10, 26]. The collaborative filtering recommendation method recommends news content based on comments and ratings from other readers/users. The reliability level of this method is good, since the rating mechanism provides pluralism. The weaknesses of recommendation systems are often scalability, change in user interest, and recency of these methods. To determine the authenticity of the news, deep learning that applies deep neural network models have also been widely used in the literature. A combination of several deep learning methods such as Long Short-Term Memory network (LSTM), Convolutional Neural Network (CNN), and Bidirectional LSTM (Bi-LSTM) was applied to a four-class label relating to news article headlines [1]. Fan et al. proposed an LSTM-based model to detect false reports in an environmental complaint system [9]. Bhattacharya et al. developed a Bi-LSTM based fake news detection model, which is an advanced version of LSTM. This model is assertive in the classification of fake news and news source detection, by using blockchain networks [4]. CNN with both different embedding models and margin loss were proposed to detect fake news with a

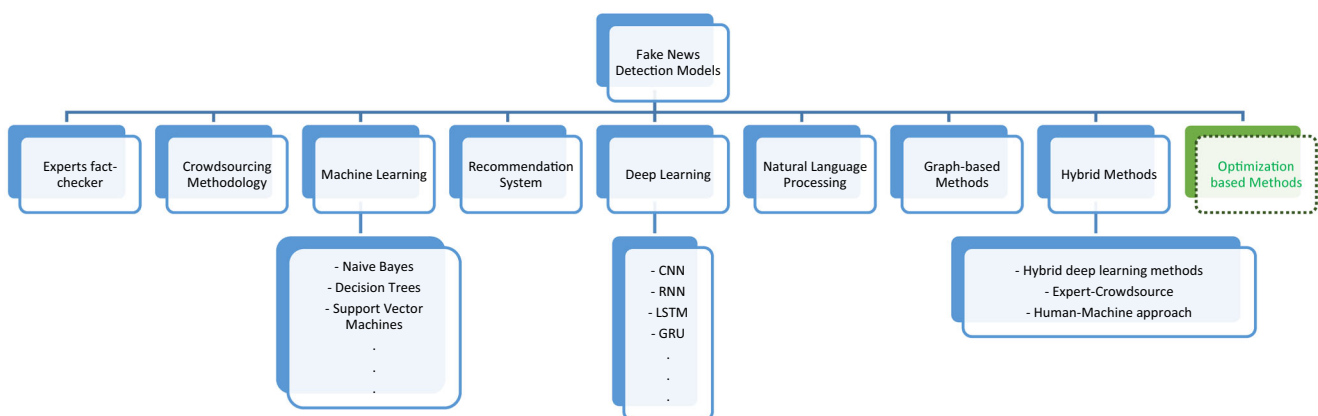


Fig. 1 The classification of different fake news detection models

higher accuracy in [12, 13]. A model based on Capsule Neural Network (CapsNet) was proposed for fake news detection in [12, 13]. Different levels of n-grams and different embedding models for news items with various lengths were applied in their work. Deep learning-based methods are capable of automatically learning latent textual representation, while capturing complex contextual patterns of news content. A detailed research on contextualized text representation and deep neural classification can be found in [38]. In addition, in fake news analysis, visual data may also be a part of the general analysis. Especially in recent years, multimodal models in which visual data are evaluated together with textual data stand out in this field [43, 44]. Besides this, different aspects and factors can be considered together when using advanced multitask deep learning models. In [20], this type of multitask model was developed, applying it to an automated fake news detection that takes into account textual anomaly and emotion factors. Raj and Meel used a deep model with two streams, which was named Coupled ConvNet [33]. This evaluates the results, obtained from a text-CNN and an Image-CNN that can use eight different deep models, together with weighted fusion. Lotfi et al. proposed a graph convolutional network-based approach that could detect rumours on Twitter conversations [22]. Another rumour detection model, which combines recurrent neural network (RNN) and Autoencoder deep models, was proposed by Chen et al. [7]. Cao et al. developed a deceptive reviews detection model that combines feature representations from the Gated Recurrent Unit (GRU), TextCNN, and Self-Attention deep models. [6]. Overall, the deep models are flexible and can adapt to complex interaction patterns. However, there are also several challenges associated with these methods, such as the volume of needed data, model complexity, and interpretability difficulties.

NLP methods work within automated detection algorithms, which involve powerful mechanisms, such as semantic and lexical analysis. Linguistic features are key factors for NLP, which can include both text style and content. Style and grammar detector with a syntactic analyser, such as Stanford parser, was reported in [18] with accurate results. These methods have limitations on the generalization of hand-crafted linguistic features across languages, topics, domains, and also in the use of the rich contextual and semantic information [40]. The graph network fake news detection model examines news content from homogeneous and heterogeneous networks [44]. Zhou and Zafarani [48] examined the graph network-based fake news detection, where the network is split into triads, communities, and nodes. The result proved to yield better results, since the method could detect fake news before spreading them. Hierarchical Graph Attention Network (HGAT) that employed the Heterogeneous Information Networks (HIN)-based fake news article in [35], with higher accuracy. Early detection in fake news analysis is an important

factor for the minimization of damages that can occur. In [41, 42] a successful propagation network-based fake news detection technique that takes this factor was developed. In summary, the graph networks can be good at early detection, but they also suffer from high computational costs, since many hyper-parameters need to be computed for these to work.

There are also hybrid methods to work well, due to both the ambiguous nature and complexity of fake news. A hybrid machine-crowd approach was proposed by [39]. The model employed the fusion of the collective effort of humans, together with that of machine learning, leading to a higher accuracy, when compared to previous studies. Hybrid deep learning models, expert-crowdsourcing, machine-crowdsourcing, and the fusion of methods from the content-based models and social context-based algorithms were also presented to use auxiliary information from different perspectives [8]. As expected, hybrid models inherit the strengths and weaknesses of all the used methods. Intelligent optimization is another methodology that can be used to detect fake news [27]. In order to obtain a better model with respect to different metrics, only one improved version of the intelligent optimization method was recently proposed [29]. In this study, the meta-heuristic approach is used. The reason is that these approaches are more explainable and suitable for parallel and hybrid work. However, these methods also have disadvantages, due to the need of sequential execution and large population management. These disadvantages may be more evident in fake news detection problems that involve multiple features. Therefore, this study proposes a different multi-thread hybrid meta-heuristic model for fake news analysis.

The next parts of the study are planned as follows; the objective of the study will be explained in Chapter 2. In Chapter 3, the basic principles of optimization-based fake news analysis are explained. The definition of the problem and the details of the proposed method are given in Chapter 4. Chapter 5 details the framework developed and the meta-heuristic algorithm used. The experiments, discussions and conclusions are presented in Chapter 6, Chapter 7 and in Chapter 8, respectively.

2 Purpose and innovative aspects of the proposed method

Deep models are more successful on complex and large data sets. On the other hand, deep models trained with relatively small data sets may not perform well. In addition, the black-box nature of machine learning and deep models makes them less explainable [45, 46]. In these two issues, metaheuristic approaches come to the fore. These approaches are both more explainable and adaptable and can be more successful in small and medium-sized data sets. However, there are two important bottlenecks in metaheuristic approaches. The first is the high

population and sequential work could increase the time and resource cost. The second is that the efficiency of exploration and exploitation decreases as the number of features of the search space increases. Hybrid metaheuristic approaches are an important technique used to overcome these challenges [11, 32]. The hybrid approaches proposed in the literature are generally based on using sequential or simultaneous joint solutions. In these approaches, the exploration and exploitation mechanisms of the metaheuristic methods proceed naturally, which causes a non-dynamic exploration process. To overcome these bottlenecks, this study proposes a multi-thread hybrid approach, which combines the performances of different meta-heuristic algorithms with today's parallel working technologies. This study introduces a new approach: the supervisor thread mechanism. The basic principle of this mechanism, which is proposed for the first time in the literature as far as is known, is to simultaneously observe and improve the performances of different meta-heuristic algorithms running in parallel in different threads. Also, the supervisor thread executes a meta-heuristic algorithm that can optimize the best values shared by other threads. The study develops a software framework based on this multi-thread model and proves its performance on different data sets. The original and innovative aspects of this study can be briefly summarized as follows:

- It could be more efficient in small and medium-sized fake news data sets than deep models.
- It proposes for the first time the use of the supervisor thread, which can observe and improve the meta-heuristic algorithm threads running in parallel.
- Furthermore, this method can also be used in general-purpose solution search strategies.
- Thanks to the supervisor thread, it enables combinations of unsearched or untested attributes in the search space for fake news detection.
- It uses the parallel-hybrid optimization technique for the first time in fake news detection.
- It separately observes both the single and multi-thread performances of different swarm-based meta-heuristic algorithms for fake news analysis.
- It offers a software framework that ensures the easy applicability of the proposed model.

3 Basic principles of optimization-based fake news detection

Optimization-based fake news analysis considers the relevant unstructured textual data set as a search space. The optimization-based method requires a binary search space. To obtain this, the relevant textual data set is pre-processed.

In the pre-processing stage, first, word roots are found by applying case conversion and some filtering operations (filtering number, N char, punctuation, etc.). Then the weights (W_i) of each word are calculated. This calculation uses the number of repetitions of each word in the data set. Thus, the weight of a word is calculated with $W_i = \frac{R_i}{R_{max}}$. Here, R_i is the number of repetitions of the i th word, and R_{max} is the maximum number of repetitions. Since the inclusion of words with very low weights in the optimization process will adversely affect performance, the search space includes words above a certain threshold value as attributes. Finally, the words included in the search space are scanned in each record in the data set. If the word is in the relevant record, it is evaluated as 1; if not, it is evaluated as 0. Thus, the dataset becomes a binary search space consisting of 1 s and 0 s.

Next, the method constructs population candidates for this search space. As an example, a swarm intelligence-based meta-heuristic algorithm has a population P containing N candidates ($P = \{\vec{X}_1, \vec{X}_2, \dots, \vec{X}_N\}$). The variables of each candidate in the population take values between $[0, 1]$ ($\vec{X}_j = \{f_1, f_2, \dots, f_K\}$, $f_i \in [0, 1]$, $K = \text{the number of the attributes}$, $i \in [1, N]$ and $i \in Z$). While calculating the fitness values of the candidates, each candidate is evaluated for each record in the edited data set. The fitness evaluations take into account two criteria. The first criterion is whether the similarity ratio between the candidate values and the related record is greater than a predefined threshold value (τ). For this, similarity functions such as Jaccard Similarity, given in Eq. 1, can be used. While performing the similarity test, the continuous value of the candidate can be used [17], or it can be converted to binary form. This study used binary representations of candidate values in similarity controls due to better performance. The second criterion is whether the class of the relevant record is the same as the candidate class. These two criteria are considered together, as shown in Table 1. Thus, the candidate can calculate the current true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) values. After this process is repeated for all records in the data set, Eq. 2 calculates the fitness value of the candidate [3]. The values of the best candidate found at the end of the iterations can provide Accuracy, Precision, and Recall metrics via (Eq. 3).

$$\text{Jaccard Value}_{\vec{x}} = \frac{\sum_{i=1}^K \text{Round}(f_i) \times \text{Record}_i}{\sum_{i=1}^K \text{Round}(f_i) + \sum_{i=1}^K \text{Record}_i - \sum_{i=0}^K \text{Round}(f_i) \times \text{Record}_i} \quad (1)$$

$$F = \frac{k_1 \times TP \times TN}{(TP + FN)(TN + FP)} + \frac{k_2 \times TP}{TP + FP} + \frac{k_3 \times TN}{TN + FN} \quad (2)$$

Table 1 Updating of TP, FP, FN, and TN for each record in the dataset

Condition	Updating
If $Jaccard\ Value_{\vec{X}} \geq \tau$ and the Class searched = = the Record Class in the data set	Increase TP by 1
If $Jaccard\ Value_{\vec{X}} \geq \tau$ and the Class searched != the Record Class in the data set	Increase FP by 1
If $Jaccard\ Value_{\vec{X}} < \tau$ and the Class searched = = the Record Class in the data set	Increase FN by 1
If $Jaccard\ Value_{\vec{X}} < \tau$ and the Class searched != the Record Class in the data set	Increase TN by 1

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

4 The problem definition and the proposed hybrid multi-thread approach

Two important issues can affect performance in optimization-based data set analysis, such as that of fake news. The first is the factors that may arise from the nature of metaheuristic algorithms, and the second is the factors that arise from the problem type or dataset structure. This section first emphasizes these effective factors, then explains the details of the Hybrid Multi-thread approach, which takes these factors into account.

4.1 The influencing factors in optimization-based data set analysis

The first factor depends on the logical mechanism used from the beginning to the end of the optimization algorithm. Metaheuristic approaches need random values by their nature. This randomness is frequently used in different steps, from the generation of the population to the candidate updates. The fact that metaheuristic methods update according to the best solution in each iteration may cause the candidate variable patterns not to change much during iterations. Here, the pattern expression is used to show what kinds of values the attributes of a candidate variables take during all iterations. The pattern averages of all candidates in a population will indicate how the search space was scanned during iterations. Using a uniform pattern structure throughout all iterations may result in indirect or direct loss of efficiency in the exploration and exploitation processes because, due to randomness, some attributes (words) may less frequently or never participate in the searching mechanism. This factor should be taken into account, especially in solving problems with many attributes, such as fake news detection. In optimization-based methods using the similarity function, the similarity values of different

candidates can easily affect the exploration process. To better understand this factor, a sample experimental study was carried out. This experimental study examined attribute patterns of two different population-based meta-heuristic algorithms used for fake news analysis. These are the Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) algorithms. The experiments ran GWO for the Covid-19 data set and PSO for the Syrian war data set. The experiments were made in single thread, and this section discusses only the attribute patterns of the populations of the algorithm.

The study adopted a different representation form for the candidates to observe the value changes of the candidate attributes. This representation form stored both the candidate values and how these candidate values changed on average in each iteration. In each iteration, the value of each attribute in the candidate is added to the previous values and then the average is calculated according to the current iteration. The average value for the attribute f_j of Candidate \vec{X}_i is calculated using Eq. 4. In this equation, CI is the current iteration number and $f_j \in \vec{X}_i$.

$$\bar{f}_j = \frac{1}{CI + 1} \sum_{k=0}^{CI} f_j^k \tag{4}$$

$$\bar{f}_j^* = \frac{1}{N} \sum_{m=1}^N \bar{f}_j^m, \quad N = population\ size \tag{5}$$

The average attribute values of the candidates also enable the calculation of the population patterns (pattern = $\{\bar{f}_1^*, \bar{f}_2^*, \dots, \bar{f}_j^*, \dots, \bar{f}_K^*\}$). This can be done with Eq. 5. Thus, on a population basis, the average changes (pattern) of the attributes can be easily seen. This method allows the easy observation of the patterns of both continuous and binary variables. Figure 2a-b show the population patterns of the sample experiments, which used binary representations of candidates. GWO and PSO were run for the classes of “true”. In the experiments, the number of iterations was 500, the population number was 20, the number of attributes for the covid data set was 135, the number of attributes for the Syrian data set was 110, and the Jaccard similarity threshold was 20%.

These graphs show the averaged values of the attributes after all iterations. Some attributes have consistently high values throughout the iterations, while some have very low values. Considering that a binary evaluation has been

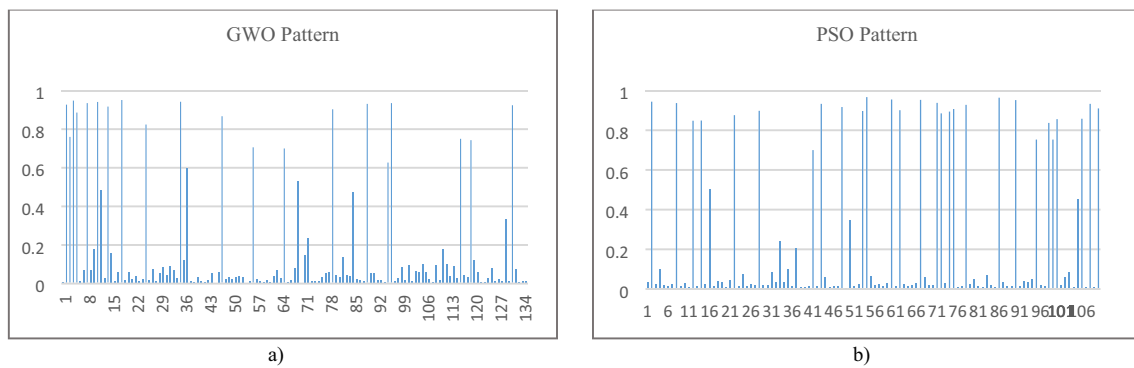


Fig. 2 GWO and PSO variable patterns obtained from demo experiments (a) GWO population patterns for Covid data set (b) PSO population patterns for Syrian war data set

completed, it can be seen that some attributes are consistently 1 and other are always 0. In fact, the patterns obtained in the experiments are an expected situation in this type of analysis. The fact that some attributes (words) are determinative requires these attributes to be in the solutions that update to the best value. Therefore, the values of some words in the pattern may be high. However, this pattern structure may reduce the efficiency of exploration and exploitation in optimization algorithms using the similarity method, because all areas of the search space and different combinations of attributes cannot be examined adequately. At certain steps during the iterations, the inclusion of new candidates with patterns different from the population pattern in the optimization can both help scan the search space more efficiently and provide flexibility against local maxima (or minima). This is one of the original points of the proposed method.

In the first factor, there is also the effect of the algorithm mechanism used. Meta-heuristic algorithms may use different logical mechanisms. For example, while PSO uses the global best mechanism, GWO has a hunting mechanism that uses both the best candidate (alpha wolf) and two other successful candidates (beta and delta wolves). Some algorithms, such as Simulating Annealing (SA), can perform a fast search with a two-candidate comparison. This enables different algorithms to obtain different results for different types of problems. On the other hand, these differences can also allow the creation of a hybrid working mechanism for solving complex problems. Another important factor is the long execution times that occur in optimizations with a large number of attributes and candidates. Execution time can increase even more for large data sets. The literature has studied the performances of evolutionary algorithms for problems with different sizes and types [21], and the parallelization technique has proven an effective solution for this. However, parallelization methods used in the literature cannot be directly applied to every problem. The type of problem and solution technique play a decisive role.

4.2 The proposed hybrid multi-thread model

This study proposes a hybrid multi-thread (HMT) solution that takes the above factors into account. With the development of today's processor technologies, multi-core and multi-thread applications have become widespread. Many programming languages can use the parallel execution offered by these technologies. This provides an ideal solution for improving the second factor (execution time) described in the previous section. The literature has proposed many multi-threading models using these advantages. These models are generally based on the principle of sharing and updating the best value [5]. This study, unlike the models in the literature, proposes a different multi-thread working model based on a supervisor thread. This supervisor thread can monitor and help the existing optimization algorithm threads running in parallel. The main mission of the supervisor thread is to provide improvement according to the first factor explained in the previous section. As far as is known, this study is the first in the literature to propose this type of model, the general structure of which is given in Fig. 3.

In multithread applications, the context switch is costly due to operations such as storing/loading registers or program counter. This cost may increase even more in context-switch operations between threads belonging to different processes. The proposed HMT model is based on the single-process architecture because of its better context switch cost. The number of threads (C) to be used in the HMT model is the total number of physical and, if available, virtual processors in a system. Apart from the main thread, HMT has three main components: Worker thread (WrT), Supervisor Thread (SvT), and Shared Object (ShO). The WrT runs a meta-heuristic algorithm. Its main task is to perform optimization-based fake news analysis. The SvT is an observer and advisor, whose task is to improve the best solutions from WrTs and produce alternative candidates according to the patterns of the WrTs. WrTs and the SvT communicate via an ShO. This is an object shared between threads, holding the best values

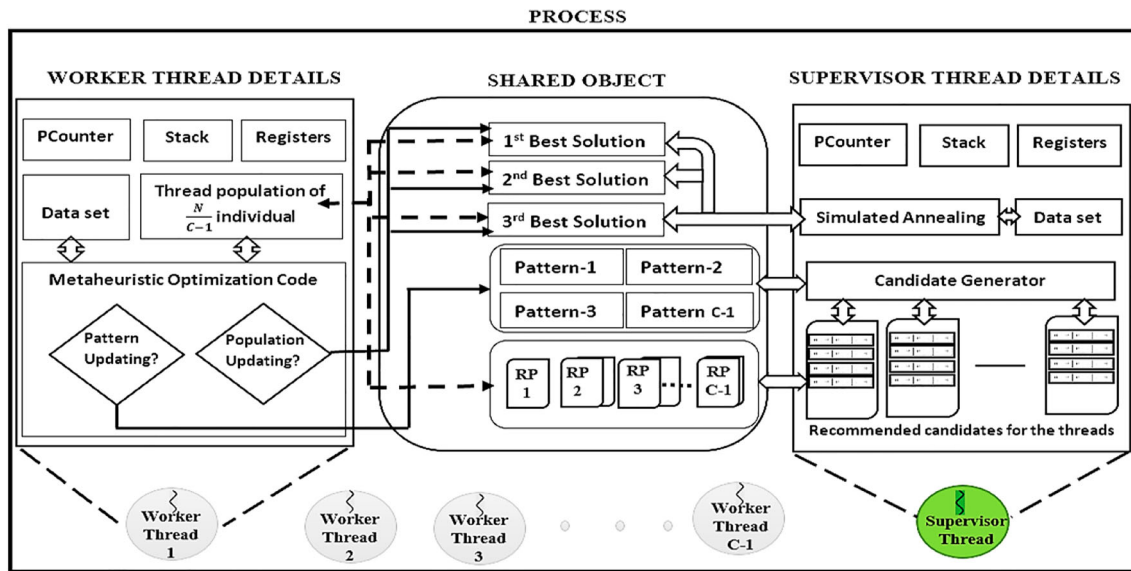


Fig. 3 The details of the HMT model

obtained during the iterations and the recommended candidates for the WrTs. The details of the HMT components can be better explained through Algorithms 1 and 2. In addition, the concepts used in the proposed method and the details of some basic knowledge can be found in Appendix 1.

Algorithm 1 shows the pseudocode of a WrT. WrTs do not share the relevant data set, but work with a copy. There are two reasons for this. First, the shared use of a data set between threads will increase the amount of blocking and cause performance loss. Second, since the dataset is in edited binary format, it is not large, and its cost for thread heap will not be high. Each WrT uses $1/(C - 1)$ of the population of the single-thread study. All WrTs compare the best values obtained from their iterations with the best values in ShO. There are three best value variables in ShO. This is because algorithms using a hunting mechanism such as GWO use more than one best value. The WrTs compare their iteration bests with the three bests in the ShO, in order. When the value in a WrT is better than one of the bests in the ShO, the corresponding value in the ShO is updated. If the best value in a WrT is not better than that in the ShO, then the candidate(s) with the best value(s) in the ShO are added to the WrT. In this case, an identical number of worst candidates is removed from the WrT population. During the iterations, WrTs calculate the patterns of both the candidates and population according to Eq. 4–5 and save them in the relevant pattern variables in the ShO. The SvT uses the pattern variables in the ShO to generate alternative candidates that will increase each WrT’s exploration ability. At certain iteration steps (S_c), WrTs include the SvT-recommended R candidates in their population and remove the worst R candidates from it. This mechanism to increase the exploration capacity of the meta-heuristic algorithm in WrT is one of the novel in HMT.

Algorithm 1 Worker Thread (WrT)

1. Get the Dataset
2. Initialize the population with $\frac{N}{C-1}$ candidates
3. Set the parameters meta-heuristic algorithm used
4. While (the iteration < MaxIteration)
5. If (iteration % $S_c=0$)
6. Add R recommended candidates to the population
7. Calculate fitness values and find the best solution(s)
8. If (the best calculated > the best(s) in the ShO)
9. Update the best(s)
10. Else
11. Get the best(s) from the ShO
12. If (population size = $\frac{N}{C-1} + R$)
13. Remove the R worst candidates from the population
14. Perform metaheuristic logic
15. Update the candidate positions
16. Update the candidates patterns by Eq.4 and 5
17. Iteration++
18. End While
19. Terminate the thread

Algorithm 2 Supervisor Thread (SvT).

1. Get the Dataset
2. While (All worker threads are running)
3. Get the best solution(s) from ShO
4. Initialize a local search for all best solutions using SA
5. If (More successful solution are find)
6. Update the best solution in ShO
7. Get the patterns from ShO
8. Clear the old recommended candidates
9. Generate R recommended candidates for each pattern
10. End While
11. Terminate the thread

Algorithm 2 is the pseudocode of SvT. Working in parallel with the WrTs, SvT has two main tasks. The first is to run the Simulated Annealing (SA) algorithm for the best values recorded in the ShO by the WrTs. SA runs on the principle of candidate-neighbour comparison and is faster than population-based algorithms. After executing SA, the best values in ShO are updated if better values have been obtained than the best values from the WrTs. This parallel searching provides two important advantages. First, it allows WrTs to reach the best values at lower iteration numbers. Second, it makes it difficult to get stuck in local maxima (or minima). The second task of SvT is to generate alternative candidates for each WrT according to the pattern information received from the WrTs. The alternative candidate generation is done to explore unscanned regions in the search space for the WrTs and to try different combinations of attributes. Thus, the effects of all the words in the fake news analysis can be seen better. Candidate generation includes analysis and punishment processes. In the analysis process, candidate generation is carried out by doing a pattern analysis for the WrT of interest. This operation is performed using Eq. 6. In this equation, x_i^R and x_i^P are the i th values ($i \in \{1, K\}$) of the recommended candidate and the pattern vector, respectively. ω_c and ω_u are chaotic and uniform random numbers, respectively ($\omega_c, \omega_u \in [0,1]$). ρ_i is the coefficient used to prevent the generation of unsuccessful similar candidates.

$$x_i^R = \begin{cases} \omega_c - \partial_L(\omega_c - 1), & \text{if } x_i^P < \partial_L \text{ and } \rho_i \omega_u > 0.5 \\ \omega_c \partial_U, & \text{if } x_i^P > \partial_U \text{ and } \rho_i \omega_u > 0.5 \end{cases} \quad (6)$$

$$\omega_c = e^{-\alpha \omega_c^2} + \beta, \quad (\alpha, \beta \in R) \quad (7)$$

The patterns are formed via iteration. Therefore, variables falling below a certain value (the condition of $x_i^P < \partial_L$) will mean that the effect and participation of the related word are generally low in the fake news analysis. In this case, the word can be given a chance to participate in the analysis with a high value. To ensure diversity in the produced candidates, value generation is carried out according to the random number ω_u . For the word to be used in the iterations, the corresponding variable value must be guaranteed to be greater than ∂_L . To provide flexibility in the HMT model, the generated value is determined by the random number ω_c , calculated using Eq. 7. Similarly, some words can be continually used throughout iterations (the condition of $x_i^P > \partial_U$). In this case, the exploration phase may always show a uniform and fixed orientation. To diversify the orientation, the effect of some active words can be reduced.

There is no guarantee that the recommended candidates generated by SvT will increase performance. It will not be beneficial to generate candidates whose patterns do not change and who do not contribute to the exploration. For this reason, different pattern variables belonging to previously generated candidates are also

created and followed by SvT. The penalty coefficient ρ_i is the determining parameter, in order not to regenerate the candidate types that do not cause a certain change in the existing pattern structure. If the similarity between the previous and new patterns in each Sc step is above a certain threshold (τ_c), a different candidate is produced by considering the candidate pattern produced in the previous step. Note that the probability of generating similar recommendation candidates is even lower when the pattern is similar in each Sc step. Therefore, the value of ρ_i is recalculated with Eq. 8, lowering the probability that a variable will be updated. Different similarity functions can be used for pattern similarity. Because of duplication sensitivity, this study calculates pattern similarity in the HMT model using the cosine similarity given in Eq. 9.

$$\rho_i = \begin{cases} e^{-\frac{\alpha}{\tau_c}}, & \text{if } \text{cosS}(x^{P_{\text{new}}}, x^{P_{\text{previous}}}) > \tau_c \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

$$\text{cosS}(x^{P_{\text{new}}}, x^{P_{\text{previous}}}) = \frac{\sum_{i=1}^K x_i^{P_{\text{new}}} x_i^{P_{\text{previous}}}}{\sqrt{\sum_{i=1}^K x_i^{P_{\text{new}}}^2} \sqrt{\sum_{i=1}^K x_i^{P_{\text{previous}}}^2}} \quad (9)$$

The ShO holds the data shared between WrTs and the SvT. The general setter and getter functions of the shared data are given in Table 2. The other functions are synchronized, except those that only transfer references. ShO also includes a termination indicator to end the application process.

5 The implementation and framework of the HMT

This section explains the implementation of the HMT model, the framework developed and the details of the metaheuristic algorithms used in the experiments.

5.1 The developed framework for HMT

The multithreading capabilities of today's object-oriented languages enable the development of flexible applications. This study, first, wrote a framework based on the HMT model. This framework was developed on the JAVA platform, a powerful object-oriented language. The framework has two important

Table 2 Shared Object Class (ShO)

Sync.	Func. Type	Function Name
Yes	Getter / Setter	BestSolution1,2 and 3
Yes/No	Getter / Setter	Pattern1,2,..C-1
Yes	Getter / Setter	RecommendedCandidates1,2,..C-1
No	Setter / Getter	Terminated

advantages. First, it easily implements and reuses different metaheuristic algorithms; second, it allows control over all thread operations. Figure 4 gives the general UML diagram of the developed framework structure. Since it is not possible to show all the variables and functions in classes in UML diagrams, only some of them are given. In the Framework, the WrT threads and SvT threads implement the *Runnable* interface. All WrT threads derive from the *AbsFake* abstract class, which contains common variables and functions. There are three important abstract methods in this class. These methods are implemented according to the meta-heuristic algorithms used in WrT classes. The SvT class only implements the *Runnable* interface. There is a composition association between the Simulated Annealing (SA) class and the SvT. The *Population* class, which enables easier use of the basic population operations, also has a composition association with all WrTs. All candidates are created from the *Candidate* class, which can be used in both population and other classes. Data resources shared between threads are kept in the *ShO* class and can be accessed by all thread classes.

5.2 The metaheuristic algorithms used in the experiments

With the framework described above, any metaheuristic algorithm can easily participate in hybrid fake news optimizations. In the experiments conducted in this study, three different swarm-based meta-heuristic algorithms were used for WrTs; Grey Wolf Optimization (GWO) [24], Particle Swarm Optimization (PSO) [15] and Dragonfly Optimization (DrO) [25] algorithms. In this section, all the details of these algorithms will not be given, but only their basic functions and principles will be explained. For a more detailed explanation about all the algorithms, the cited references can be consulted.

GWO is a swarm-based metaheuristic algorithm that use a hunting mechanism. Inspired by *Canis lupus* wolves, this algorithm has four types of wolves with a hierarchical structure.

The top of the hierarchy is the alpha (α) wolf, which has the best value; and next it comes the beta (β) and delta (δ) wolves, in this order. The last wolf type, omega (w), follows these three leader wolves. The general expression of the tracking mechanism is expressed with Eq. 10.

$$\vec{X}(i+1) = \vec{X}_p(i) - \vec{A} \circ \vec{D} \tag{10}$$

In this equation, \vec{A} is calculated by $\vec{A} = 2\vec{a} \circ \vec{v}_1 - \vec{a}$, which depends on linearly decreasing coefficient \vec{a} , and random vector \vec{v}_1 . The random vector $\vec{C} = 2\vec{v}_2$ is effective in the computation of the \vec{D} , being computed separately for each leader wolf by Eq. 11. The next position for the chasing wolves is determined by Eqs. 12 and 13.

$$\begin{aligned} \vec{D}_\alpha &= \left| \vec{C}_1 \circ \vec{X}_\alpha - \vec{X} \right| & \vec{D}_\beta &= \left| \vec{C}_2 \circ \vec{X}_\beta - \vec{X} \right| & \vec{D}_\delta &= \left| \vec{C}_3 \circ \vec{X}_\delta - \vec{X} \right| \end{aligned} \tag{11}$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \circ \vec{D}_\alpha & \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \circ \vec{D}_\beta & \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \circ \vec{D}_\delta \end{aligned} \tag{12}$$

$$\vec{X}(i+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{13}$$

PSO is one of the most well-known swarm-based algorithms. The basic principle of the algorithm is that in each iteration, all the candidates determine their position according to the best of the population, as well as their parameters. In ($i + 1$)th iteration, the velocity vector (\vec{V}_{i+1}^k) is decisive in determining the new position of the k th candidate, being calculated with Eqs. 14 and 15. In this equation, \vec{V}_i^k represents the current velocity, w_p the inertia coefficient, c_1, c_2 the self and

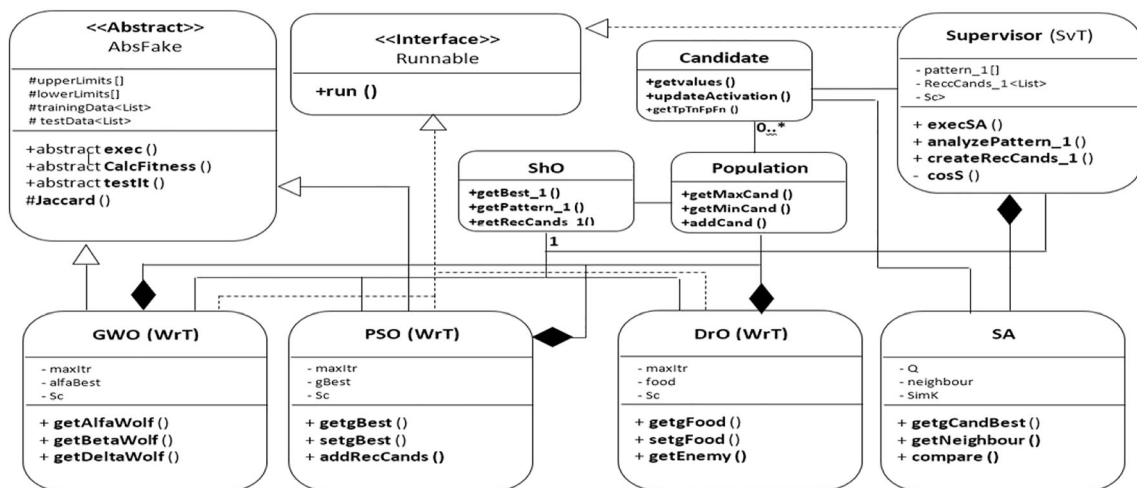


Fig. 4 The executable framework UML diagram of the HMT

swarm confidence coefficients, respectively, and \vec{x}_i^k the candidate positions. The velocity depends on the best swarm (\vec{g}_{best}) and the best particle (p_b^k).

$$\vec{V}_{i+1}^k = w_p * \vec{V}_i^k + c_1 * rand * (p_b^k - \vec{x}_i^k) + c_2 * rand * (\vec{g}_{best} - \vec{x}_i^k) \quad (14)$$

$$\vec{x}_{i+1}^k = \vec{x}_i^k + \vec{V}_{i+1}^k \quad (15)$$

The DrO is an algorithm based on the dragonfly's movements and hunting methods. Dragonflies have different movement controls within the swarm, which are the separation (\vec{S}_i), alignment (\vec{A}_i), cohesion (\vec{C}_i), attraction (\vec{F}_i), and distraction (\vec{E}_i). The DrO algorithm uses an iterative neighborhood mechanism, and in this way, the vectors \vec{S}_i , \vec{A}_i and \vec{C}_i are computed according to the neighborhood information. \vec{S}_i represents the collision avoidance for neighboring candidates, being determined according to the position of the candidate \vec{X} and neighbor \vec{X}_j . \vec{A} is calculated by averaging the velocities (\vec{V}_i) of neighboring candidates, while \vec{C}_i represents the motion consistency of neighboring candidates. The mathematical models of these three systems that use neighborhood information are as in Eq. 16.

$$\vec{S}_i = -\sum_{j=1}^N \vec{X} - \vec{X}_j, \quad \vec{A}_i = \frac{\sum_{j=1}^N \vec{V}_j}{N}, \quad \vec{C}_i = \frac{\sum_{j=1}^N \vec{X}_j - \vec{X}}{N} \quad (16)$$

\vec{F}_i and \vec{E}_i determine the course of action for food and the enemy, respectively, and their general expressions are $\vec{F}_i = X^+ - X^-$ and $\vec{E}_i = X^+ + X^-$. Here, the food and enemy represent the best value (X^+) and the worst value (X^-), respectively. The new positions of the candidates are computed with a certain amount of steps (ΔX). In this step, each operation type has a certain weight coefficient (s, a, c, f, and e). The inertia coefficient (w) of the candidate is effective in determining ΔX . Thus, a candidate's new position is found by using Eqs. 17 and 18.

$$\Delta \vec{X}_{i+1} = (s \vec{S}_i + a \vec{A}_i + c \vec{C}_i + f \vec{F}_i + e \vec{E}_i) + w \Delta \vec{X}_i \quad (17)$$

$$\vec{X}_{i+1} = \vec{X}_i + \Delta \vec{X}_{i+1} \quad (18)$$

The SA algorithm is run by supervisor thread (SvT) to improve the best values of WrTs. SA algorithm is a stochastic and fast algorithm. Using a solution candidate (\vec{X}) and a neighbor candidate (\vec{X}_N), the algorithm makes a comparison between them. The solution candidate for the next iteration is one with the best value, with the neighborhood vector (\vec{N}) being used for the computation of the new neighbor candidate. The probability acceptance (Pa) value is important so the neighbor candidate value can be accepted as the solution, with

Eq. 19 being used in this process. Here, Q is the current temperature and $\Delta Z = fitness(\vec{X}) - fitness(\vec{X}_N)$.

$$Pa = \begin{cases} 1, & \text{if } \Delta Z < 0 \\ e^{-\frac{\Delta Z}{Q}}, & \text{if } \Delta Z \geq 0 \end{cases} \quad (19)$$

The mathematical and logical operations of the GWO, PSO, and DrO are executed on lines 14 and 15 of Algorithm 1. This algorithm compares their best with the best in the ShO in each iteration and, if necessary, uses the best(s) from the ShO. The population management is also carried out by the WrTs.

6 Experiments

In this section, the details of the experiments carried out in the study will be given. In addition, the training step performances of all the optimization-based methods will be presented in this section. Regarding the comparative evaluation of the results obtained from the experiments, this will be discussed in the next section. Three different datasets were used in the experiments, which were the Covid-19 [19], Syrain [37], and general-news [2] datasets. The Covid-19 is a dataset compiled from news related to the pandemic, containing a total of 3119 records. Of these, 1058 were classified as fake and 2061 as true. The dataset created from the news of the Syrian war has a total of 804 records, including 378 fake news and 426 true news. The last dataset was compiled from political news, having a total of 44,858 records, of which 23,441 were labeled as fake and 21,417 as true. All the datasets have been brought into a binary form, by the methods described in Section 3 for optimization-based analysis and the threshold for joining the search space was set to 10%. Accordingly, 135 (134 attributes +1 class) in the Covid dataset, 110 in the Syrian dataset, and 137 in the general-news dataset formed the different search spaces. 70% of the data was used for training and 30% for testing. General information about the datasets used in the experiments is given in Table 3.

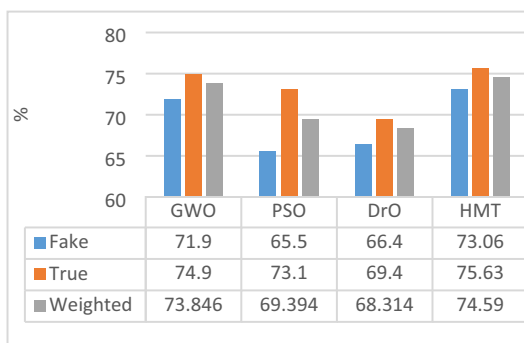
During the experiments, first each optimization algorithm was run as a single thread. Afterwards, algorithm performances could be observed separately. Finally, multi-thread experiments using the HMT model were carried out. For a more comprehensive evaluation, all these optimization-based fake news analysis results were compared with the results of 9 different standard classification algorithms, as well as 5 well-known deep models (CNN, RNN, GRU, LSTM, and Bi-LSTM). The hyper-parameters used for CNN were: number of layer equals to 3, channel size of 6, kernel size of (3,3), stride and padding of (1,1), pooling of (2,2), dropout equals to 0.25, and FFN layer numbers of 2. For RNN, LSTM, and GRU, the used number of layers was 3 and the chosen dropout was 0.25. FFN included 4 layers with 16 nodes, having used the Adam optimizer. Each experiment was run a total of 20 times, so the statistical performances of the proposed model

Table 3 The properties of the data sets used in the experiments

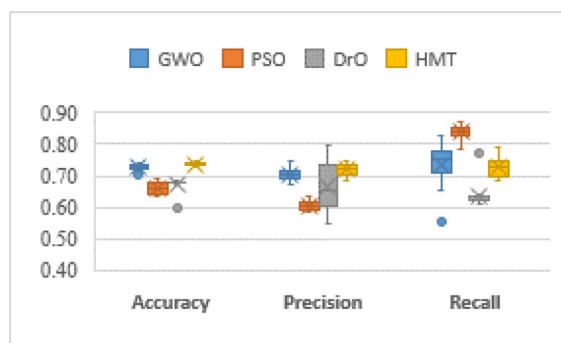
Data set	Total Records	Training / Test Splitting Rate (%)	Attribute Words	Training data set size	Test data set size	Class
<i>Covid-19</i>	3119	70 / 30	134	2183×135 (134+1)	936×135	True/Fake
<i>Syrian</i>	804	70 / 30	109	563×110 (109+1)	241×110	True/Fake
<i>General-news</i>	44,858	70 / 30	136	31,401×137 (136+1)	13,457×137	True/Fake

could be analyzed. The number of iterations in all the optimization methods was 500. The analysis also examine which words are effective in classifying for a more explainable evaluation. For this, the method considers the statistical evaluation of attribute weights. The solution candidates of the algorithms take a weight between [0,1] for each attribute, and the binary equivalents of these weights determine the position of the

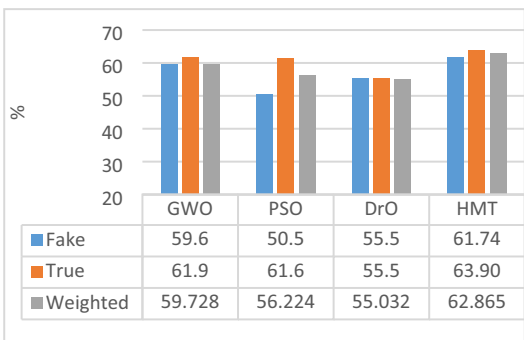
candidate in the search space. Since the binary 1 value plays a decisive role in the Jaccard similarity, the weights of the attributes give an idea about which words are more effective in determining the class. Therefore, the average of the attribute weights of the best solutions obtained in all experiments is used for word effectiveness analysis. In the M experiment, the average efficiency value of the *i*th feature for class C is



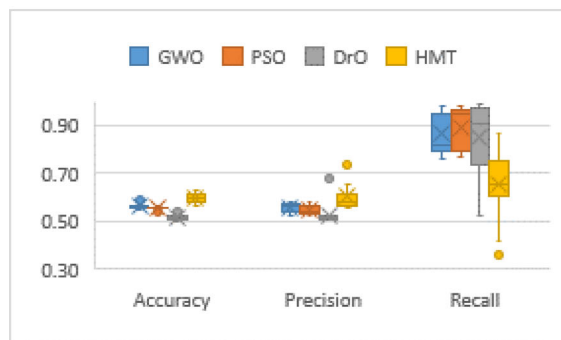
a)



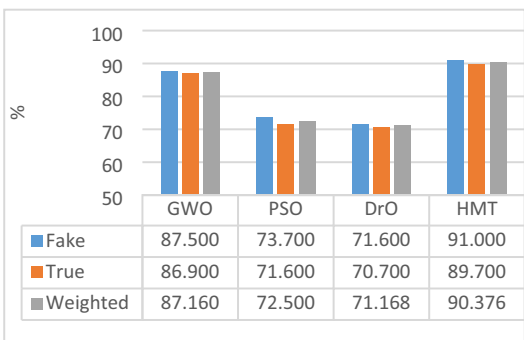
b)



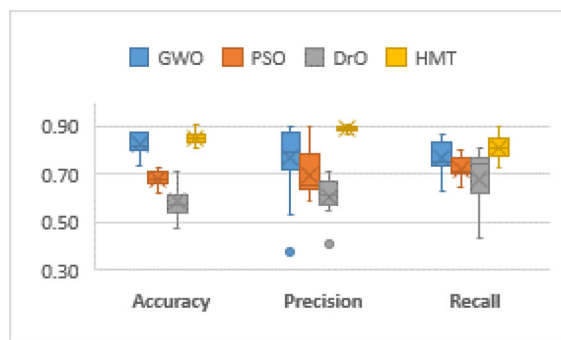
c)



d)



e)



f)

Fig. 5 All results from the training stages for all the data sets (a) The best accuracy scores for Covid-19 data set (b) All metric results for Covid-19 data set (c) The best accuracy scores for the Syrian war data set (d) All

metric results for Syrian war data set (e) The best accuracy scores for the general-news (f) All metric results for the general-news

found with Eq. 20, where B^k represents the best solution in the experiment k.

$$f_i^C = \frac{1}{M} \sum_{k=1}^M w_i^{B^k}, \quad i \in \text{Attributes}, \quad C \in \{\text{True}, \text{False}\} \tag{20}$$

The experiments were carried out by four threads on a platform with a 4-core Intel i5-4570 (no HT) processor and 16 GB of memory. Optimization-based methods consist of two steps: training and test. Before discussing the comparative results, it will be useful to give the training performances of the optimization-based methods. In the article, the results will be given in [Accuracy, Precision, Recall] format and in a normalized form.

Figure 5 shows the results from all the training experiments for the three datasets. Although the basic metric is accuracy, the precision and recall performances of the methods are also monitored to improve the analysis. Figure 5a shows the highest accuracy results obtained in the training experiments for the Covid data set. The highest accuracy values were obtained by the HMT and GWO, in this order. Considering the weighted values, the HMT has the highest score with 74.59%, while the lowest score is the DrO with 68.31%. For the general-news dataset, the weighted scores according to the results obtained in all training experiments for True and Fake classes are shown in Fig.

Table 4 Test results according to the best accuracy score for the Covid-19 data set

Deg	Method	Acc	Pre	Rec
1	SVM (Support vector mechine)	78.8	0.785	0.788
2	HMT	74.31	0.738	0.733
3	Bi-LSTM	74.04	0.734	0.74
4	JRIP	73.7	0.726	0.737
5	FilteredClassifier	73.6	0.731	0.736
6	GWO	72.91	0.716	0.698
7	LSTM	71.474	0.706	0.715
8	Ridor	71.4	0.702	0.715
9	RNN	70.83	0.698	0.708
10	Ibk	70.4	0.717	0.704
11	CNN	70.19	0.692	0.702
12	DT (Decision Tree)	69.9	0.698	0.7
13	NB (Naive Bayes)	68.9	0.723	0.689
14	GRU	68.59	0.668	0.686
15	DrO	68.54	0.458	0.641
16	PSO	68.46	0.619	0.774
17	One	67.9	0.651	0.679
18	FFN (Feedfoward neural network)	67.31	0.661	0.673
19	RandomTree	66.4	0.662	0.665

The bold expressions show the results of the proposed and used methods in the article

Table 5 All results for Fake and True classes

GWO	PSO						DrO						HMT											
	True			Fake			True			Fake			True			Fake								
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec						
Best	71.500	0.638	0.569	74.700	0.793	0.961	63.200	0.464	0.464	67.000	0.715	0.981	66.300	0.750	0.412	69.900	0.700	0.723	0.494	75.000	0.762	0.966		
Worst	64.500	0.476	0.082	71.600	0.714	0.783	43.800	0.327	0.327	42.100	0.686	0.940	40.400	0.222	0.003	67.800	0.682	0.921	68.056	0.551	0.230	71.261	0.707	0.905
Mean	68.799	0.567	0.385	73.215	0.749	0.898	52.095	0.374	0.374	56.945	0.696	0.959	64.665	0.491	0.028	68.420	0.689	0.949	70.730	0.651	0.312	73.130	0.731	0.939
Median	69.250	0.574	0.392	73.400	0.751	0.907	52.550	0.375	0.375	58.8	0.694	0.957	65.950	0.523	0.006	68.300	0.689	0.949	70.459	0.649	0.303	73.130	0.728	0.943
Std	1.833	0.042	0.097	0.788	0.022	0.051	5.819	0.037	0.037	0.066	0.830	0.007	0.949	0.192	0.088	1.202	0.097	0.013	1.350	0.052	0.061	0.998	0.014	0.019

Table 6 The weighted results for all methods

	GWO - weighted			PSO - weighted			DrO- weighted			HMT- weighted		
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec
Best	72.910	0.722	0.821	68.464	0.619	0.875	68.540	0.712	0.769	74.310	0.749	0.798
Worst	69.798	0.641	0.647	61.244	0.567	0.774	58.550	0.529	0.610	71.017	0.666	0.681
Mean	71.713	0.687	0.724	63.662	0.586	0.827	67.143	0.622	0.636	72.314	0.704	0.726
Median	71.760	0.690	0.712	63.473	0.584	0.832	67.482	0.635	0.632	72.208	0.701	0.724
Std	0.817	0.023	0.049	2.056	0.014	0.024	2.005	0.067	0.032	0.951	0.024	0.026

5b. As seen in the figure, the value changes in precision and recall metrics are different from the accuracy metric, since while in the HMT and DrO the precision metric were better, the best values were captured by the PSO in the recall metric. As can be seen in Fig. 5c, the best accuracy values were also obtained by the HMT in the training experiments for the Syrian war. This success is followed by single-thread GWO, PSO, and DrO, in this order. Looking at all the weighted results, HMT is more successful in the accuracy and precision metrics, while single thread methods are more successful in the recall. In the training experiments for the general-news dataset, the best performance in all metrics was obtained when using HMT. The order of performance in single-thread methods was not changed for this dataset.

7 The comparative results and discussion

In this section, the results of the experiments are discussed in comparison with the results of 15 different standard classification algorithms. The test results have been evaluated according to the highest accuracy performances, both the statistics of the class and the weighted scores, and also the Friedman tests. In addition to this, it will also be examined in this section how the proposed method mechanisms affect the optimization processes.

7.1 The Covid-19 data set results

The best candidates in the training stage for the Covid dataset were used for the test data in the second phase. Table 4 shows the comparative accuracy results. The ranking criterion is accuracy, and the weighted values of optimization-based approaches are used for the evaluation.

The detailed statistical results for all the experiments are given in Tables 5 and 6. The HMT achieved the second-best accuracy, while the single-thread GWO achieved the sixth-best. The best accuracy value in the Covid dataset was captured by the SVM. DrO and PSO accuracy performances were below Naïve Bayes (NB) and Decision Tree (DT). Generally, no big differences were observed between test and training results. Standard deep models were not very successful for this medium-sized dataset, with the most successful model being the Bi-LSTM. At the Precision value, the HMT's best candidate is [74.24, 0.749, 0.706]. Among the standard classification algorithms, the best Precision values belong to SVM, with the following values [78.80, 0.785, 0.788]. The top three results in the Recall metric were obtained by optimization-based approaches, more specifically PSO-[61,788, 0.574, 0.875], GWO-[69.798, 0.643, 0.821] and HMT-[71,733, 0.666, 0.798]. According to the statistical results, both the mean and median values of HMT are very close to each other, while the standard deviation being small. In addition, as can be seen from Table 7, there is a significant difference in the Friedman tests for all the classes.

Both SvT and WrTs can also report how the mechanisms proposed in the HMT affect the optimization processes. Consequently, it can be observed which mechanism is more effective in determining the optimum values. Figure 6a shows the average number of times SA and candidate recommendation mechanisms are effective in updating the best values across all experiments. Accordingly, it is seen that the pattern-based candidate recommendation system is more effective than SA, with the most updated value being the third best value. The SA was more effective in the first 30 iterations.

Table 7 The results of the Friedman tests for the Covid-19 experiments

	Sig.Level	p value for accuracy	Significant for accuracy	p value for precision	Significant for precision	p value for recall	Significant for the recall
Fake	0.05	< 1E-3	Yes	< 1E-3	Yes	< 1E-3	Yes
True	0.05	< 1E-3	Yes	< 1E-3	Yes	< 1E-3	Yes
Weighted	0.05	< 1E-3	Yes	< 1E-3	Yes	< 1E-3	Yes

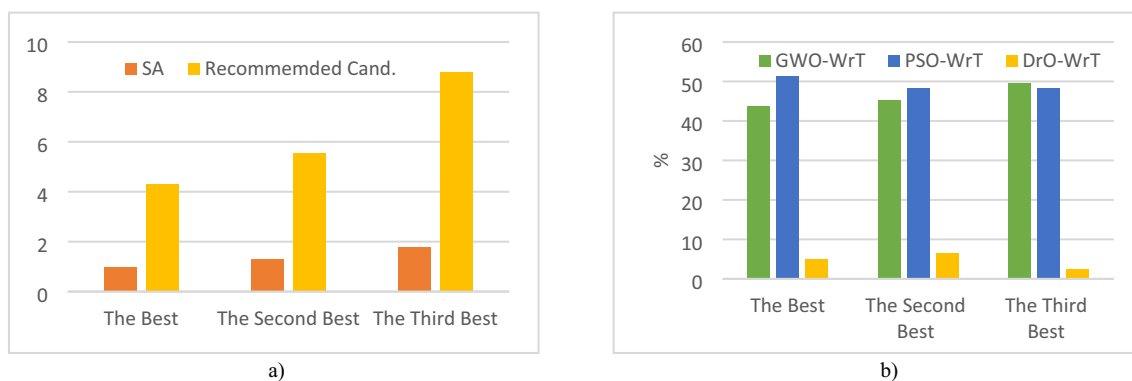


Fig. 6 Effects of mechanisms and algorithms used in HMT on the best values in the ShO (a) Average numbers of influences used mechanisms in the HMT (b) Best values update rates for WrTs

Figure 6b shows the average update rates of the WrTs on the best values of the ShO. Interestingly, although GWO was more successful in single-thread experiments, PSO updated the best values in a more efficient way than GWO when using the HMT model. One of the most important reasons for this is that the PSO algorithm is faster than GWO, rather than the dynamism of PSO. It is also important to state that the effect of DrO was quite low compared to the others.

Figure 7 shows the 10 most effective word stems and average weights among 135 word stems for True and Fake analysis in the Covid experiments. In the experiments conducted for both classes, the words “corona” and “virus” with the highest frequency had the highest weights. On the other hand, words such as “quarantine(e)”, “infect”, “report” and “travel” often had high weight values in the best solutions in True class analyses. In fake analyses, words such as “flu”, “pandem(ia)”, “outbreak” and “develop” stepped forward.

7.2 The Syrian war data set results

The optimization-based methods provided a clear superiority in the experiments with this dataset. The results for the test data set are given in Table 8 regarding the accuracy values. The HMT candidate achieved the best values in both accuracy and precision

metrics, followed by the LSTM and RNN. The PSO’s candidates in the training stages fell behind in the test data, with the standard classification algorithms being behind standard deep models. Among the standard methods, the NB and SVM came to the fore, and the HMT achieved a more effective success in this dataset. The best precision belongs the candidate of HMT with [0.539, 0.580, 0.407]. As can be seen from the statistical results in Tables 9 and 10, both mean-median closeness, and small standard deviation indicate that the HMT keeps its stability. The Friedman test results are presented in Table 11. Accordingly, it is seen that there is a significant difference between the results. The average effects of the mechanisms used in the SvT are shared in Fig. 8, showing that the effect of the SA was again limited, occurring only during the initial iterations. The SA mostly updated the second and third best values. However, the candidate recommendation mechanism was more effective than the SA in updating the best values and improving the performance of HMT. In parallel working, considering the average update performances of the WrTs, it is seen that while the GWO and PSO updated at approximately the same rate, the DrO lagged behind the others.

The most effective 10 words and their weights in the true and fake Syrian war news analyses are shown in Fig. 9. In both classes, the words “Syrian” and “kill” were the words that have

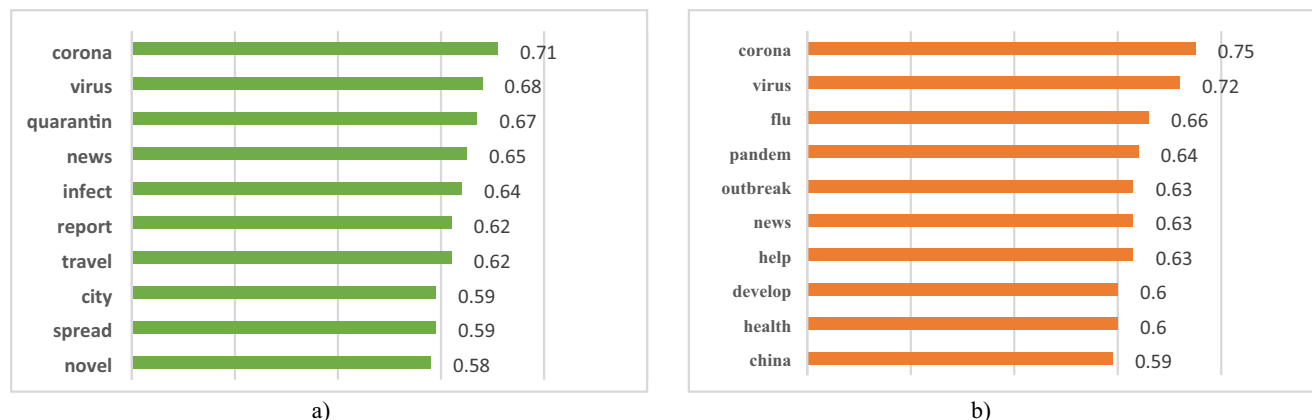


Fig. 7 The ten word stems with the highest average weights in the Covid data set experiments (a) True (b) Fake

Table 8 Test results according to the best accuracy scores for the Syrian data set

Deg	Metot	Acc	Pre	Rec
1	HMT	57.47	0.575	0.562
2	LSTM	56.79	0.573	0.568
3	RNN	55.56	0.563	0.556
4	GWO	54.32	0.528	0.765
5	GRU.	54.32	0.542	0.543
6	FFN	54.32	0.559	0.543
7	DrO	54.18	0.536	0.602
8	NB	53.9	0.538	0.539
9	SVM	53.5	0.533	0.535
10	DT	52.7	0.528	0.527
11	JRIP	51.9	–	0.519
12	FilteredClassifier	51.9	–	0.519
13	PSO	50.93	0.505	0.976
14	Bi-LSTM	50.62	0.524	0.506
15	RandomTree	50.2	0.5	0.502
16	One	49	0.403	0.49
17	CNN	48.15	0.488	0.481
18	Ibk	47.7	0.477	0.477
19	Ridor	47.3	0.372	0.473

The bold expressions show the results of the proposed and used methods in the article

the highest effect on the similarity rate in the candidate solutions. However, words that show news sources, such as “reuter”, were also frequently used in True analysis. Words such as “terrorist”, “suicid(e)” and “province” were effective in fake news analysis.

7.3 The general-news data set results

Table 12 shows the best accuracy values from the general-news test data and the results of the standard classification algorithms and deep models comparatively. The most successful result for this dataset was obtained by the deep models. Also, SVM and JRIP were successful in this dataset. Although HMT’s performance was high, it was sixth in this dataset, with single-thread approaches not being successful in this dataset. The results also show that in fake news detection, the performance of deep models trained with large datasets is better than the models trained by small-medium datasets. In other words, metaheuristic approaches could fill this gap. As can be seen from Tables 13 and 14, the HMT showed a more stable behavior with best-worst and mean-median intervals, when compared to other methods. In addition, it was observed that there were also significant differences between the results in the Friedman tests, the results of which are given in Table 15.

When the values given in Fig. 10 are looked at, according to the previous datasets, the SA mechanism of HMT has the highest rate of influence in this data set. Its effect has increased,

Table 9 All results of the Fake and True classes for the Syrian data set

	GWO						PSO						DrO						HMT					
	True			Fake			True			Fake			True			Fake			True			Fake		
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec			
Best	53.300	0.504	0.982	55.400	0.579	0.797	47.900	0.474	0.974	55.400	0.579	0.797	55.000	0.537	0.982	54.100	0.552	0.992	56.612	0.615	0.921	58.264	0.576	0.898
Worst	45.500	0.436	0.316	44.600	0.474	0.430	45.900	0.464	0.956	44.600	0.474	0.430	45.900	0.434	0.009	47.100	0.500	0.320	47.107	0.400	0.018	49.174	0.514	0.563
Mean	47.838	0.471	0.845	51.520	0.537	0.602	46.900	0.469	0.968	51.520	0.537	0.602	47.955	0.475	0.840	52.930	0.535	0.846	52.005	0.493	0.468	53.059	0.542	0.737
Median	47.100	0.470	0.956	52.100	0.540	0.617	47.100	0.470	0.974	52.100	0.540	0.617	47.100	0.470	0.982	53.300	0.533	0.961	52.273	0.490	0.500	52.479	0.537	0.723
Std	2.326	0.014	0.209	3.023	0.027	0.100	0.482	0.241	0.008	0.030	0.027	0.100	10.029	0.021	0.289	1.418	0.209	0.204	2.795	0.048	0.225	0.023	0.018	0.098

Table 10 The weighted results for all methods

	GWO - weighted			PSO - weighted			DrO- weighted			HMT- weighted		
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec
Best	54.184	0.536	0.882	50.932	0.507	0.983	54.324	0.533	0.987	57.471	0.580	0.909
Worst	45.224	0.469	0.489	49.024	0.496	0.853	46.908	0.484	0.512	49.686	0.478	0.346
Mean	49.753	0.506	0.719	49.854	0.501	0.930	50.542	0.506	0.843	53.553	0.519	0.608
Median	49.732	0.505	0.741	50.004	0.501	0.949	50.324	0.503	0.902	53.223	0.512	0.619
Std	2.104	0.016	0.104	0.561	0.003	0.043	1.542	0.011	0.147	2.159	0.028	0.125

Table 11 The results of the Friedman tests for the Syrian data set

	Sig.Level	p value for accuracy	Significant for accuracy	p value for precision	Significant for precision	p value for recall	Significant for the recall
Fake	0.05	< 1E-3	Yes	0.044	Yes	< 1E-3	Yes
True	0.05	< 1E-3	Yes	0.029	Yes	< 1E-3	Yes
Weighted	0.05	< 1E-3	Yes	0.018	Yes	< 1E-3	Yes

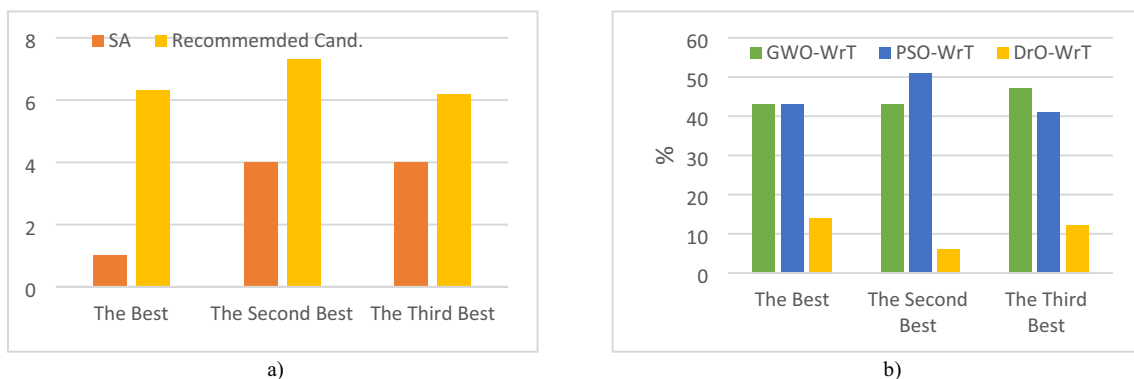
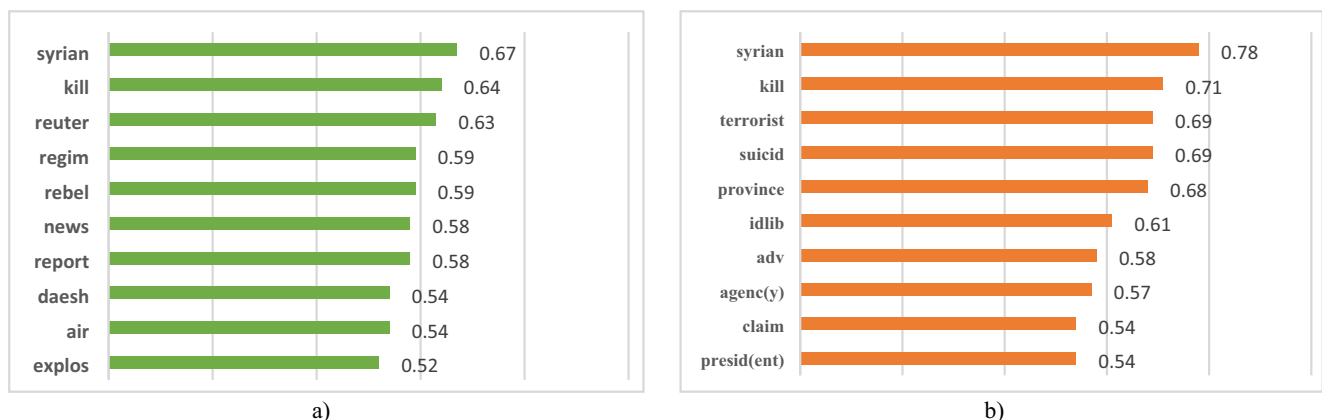
**Fig. 8** Effects of mechanisms and algorithms used in HMT on the best values in the ShO in the Syrian war dataset experiments. **a** Average numbers of influences used mechanisms in the HMT **b** Best value update rates for WrTs**Fig. 9** The ten word stems with the highest average weights in the Syrian war data set experiments **(a)** True **(b)** Fake

Table 12 Test results according to the best accuracy score for the general-news data set

Deg	Method	Acc	Pre	Rec
1	LSTM	94.31	0.946	0.943
2	SVM	93.5	0.935	0.935
3	CNN	91.82	0.918	0.918
4	JRIP	91.2	0.913	0.912
5	Bi-LSTM	90.57	0.916	0.906
6	HMT	90.18	0.903	0.902
7	GRU.	90.17	0.905	0.901
8	DT	90.1	0.901	0.901
9	RNN	89.38	0.895	0.893
10	Ibk	89.2	0.892	0.892
11	FFN	88.12	0.901	0.846
12	GWO	87.4	0.896	0.834
13	Ridor	87.32	0.869	0.864
14	FilteredClassifier	86.9	0.893	0.824
15	NB	86.7	0.869	0.867
16	RandomTree	86.4	0.864	0.864
17	PSO	71.61	0.708	0.779
18	DrO	71.39	0.708	0.771
19	One	68.1	0.787	0.681

The bold expressions show the results of the proposed and used methods in the article

especially when updating the second and third best values. However, the candidate recommendation mechanism had a similar effect when compared to the other datasets. On the other hand, the effect of PSO-WrT had a higher increase, which may be due to the fact that the general news dataset is bigger than the other datasets. The fact that the PSO is faster than the GWO has helped it to perform more updates in parallel mode. Again, the DrO was not very successful in the update competition.

As a result, the HMT model has been more successful than single-thread optimization algorithms. At the same time, it has been seen that it achieves very competitive and successful performances, when compared to other standard methods in the literature. In addition, it can produce better results for small and medium-sized datasets than deep models. When all metrics are considered together, the results of the HMT and optimization-based approaches generally have outperformed the results of standard classification methods. Considering the performance of the optimization-based approaches among themselves, the GWO that uses a multiple leader hunting mechanism is more successful compared to the PSO and DrO using single the gbest (or food). On the other hand, the most successful standard classification algorithm was the SVM, followed by the JRIP and DT.

Figure 11 shows the effective words and their average weights in the analysis that was made with the general news dataset. The word stems representing the news source or official institution (“reuter”, “minist”, “senat”, etc.) were decisive

Table 13 All results of the Fake and True classes for the general-news data set

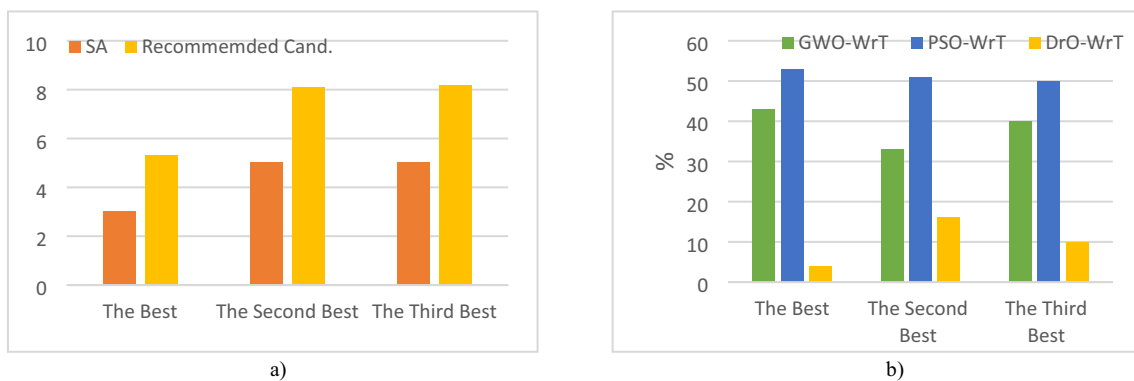
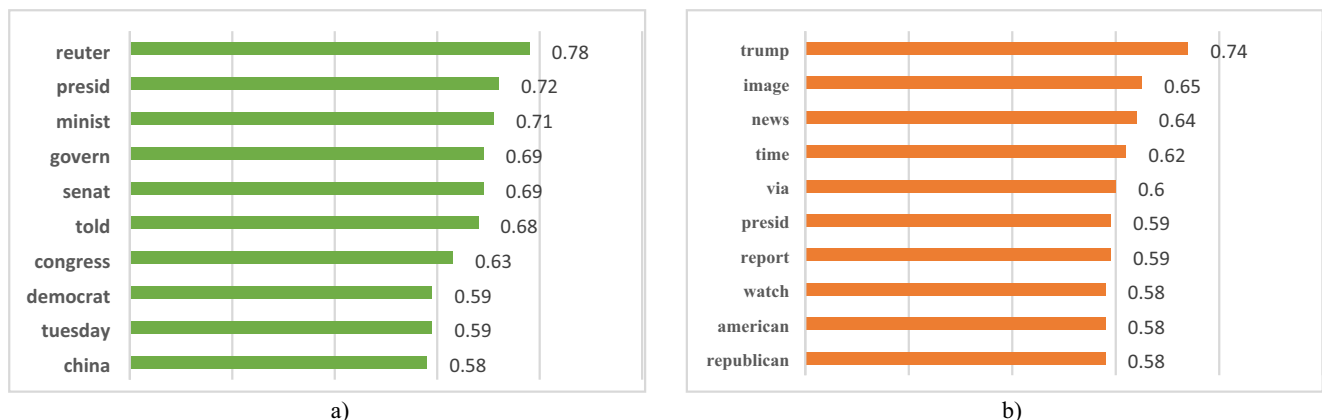
	GWO			PSO			DrO			HMT								
	True			Fake			True			Fake								
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec						
Best	87.500	0.896	0.865	87.500	0.896	0.865	71.200	0.712	0.778	71.600	0.886	0.838	89.600	0.920	0.896	90.800	0.914	0.908
Worst	73.500	0.212	0.625	73.500	0.212	0.607	46.900	0.120	0.051	30.000	0.539	0.265	80.200	0.863	0.723	80.400	0.853	0.728
Mean	81.710	0.591	0.751	83.835	0.715	0.790	58.410	0.471	0.630	58.370	0.614	0.723	83.062	0.896	0.765	85.118	0.884	0.809
Median	87.100	0.872	0.834	86.400	0.863	0.834	55.900	0.556	0.730	58.700	0.552	0.749	82.709	0.896	0.750	85.126	0.886	0.806
Std	6.315	0.330	0.100	5.182	0.282	0.091	7.717	0.233	0.218	10.660	0.111	0.157	2.382	0.151	0.048	0.027	0.020	0.054

Table 14 The weighted results for all methods

	GWO - weighted			PSO - weighted			DrO- weighted			HMT- weighted		
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec
Best	87.448	0.896	0.865	71.600	0.807	0.801	71.392	0.712	0.807	90.176	0.908	0.902
Worst	73.500	0.212	0.625	61.720	0.174	0.641	46.796	0.328	0.379	80.868	0.867	0.736
Mean	82.730	0.651	0.770	67.538	0.581	0.725	58.391	0.540	0.675	84.049	0.891	0.786
Median	82.076	0.576	0.747	67.546	0.656	0.716	57.244	0.553	0.740	83.909	0.891	0.782
Std	4.619	0.238	0.074	3.030	0.171	0.047	7.213	0.125	0.146	2.000	0.010	0.036

Table 15 The results of the Friedman tests for the general-news data set

	Sig.Level	p value for accuracy	Significant for accuracy	p value for precision	Significant for precision	p value for recall	Significant for the recall
Fake	0.05	< 1E-3	Yes	< 1E-3	Yes	< 1E-3	Yes
True	0.05	< 1E-3	Yes	< 1E-3	Yes	< 1E-3	Yes
Weighted	0.05	< 1E-3	Yes	< 1E-3	Yes	< 1E-3	Yes

**Fig. 10** Effects of mechanisms and algorithms used in HMT on the best values in the ShO, in the general-news dataset experiments. **a** Average numbers of influences used mechanisms in the HMT **b** Best value update rates for WrTs**Fig. 11** The ten word stems with the highest average weights in the general news data set experiments **(a)** True **(b)** Fake

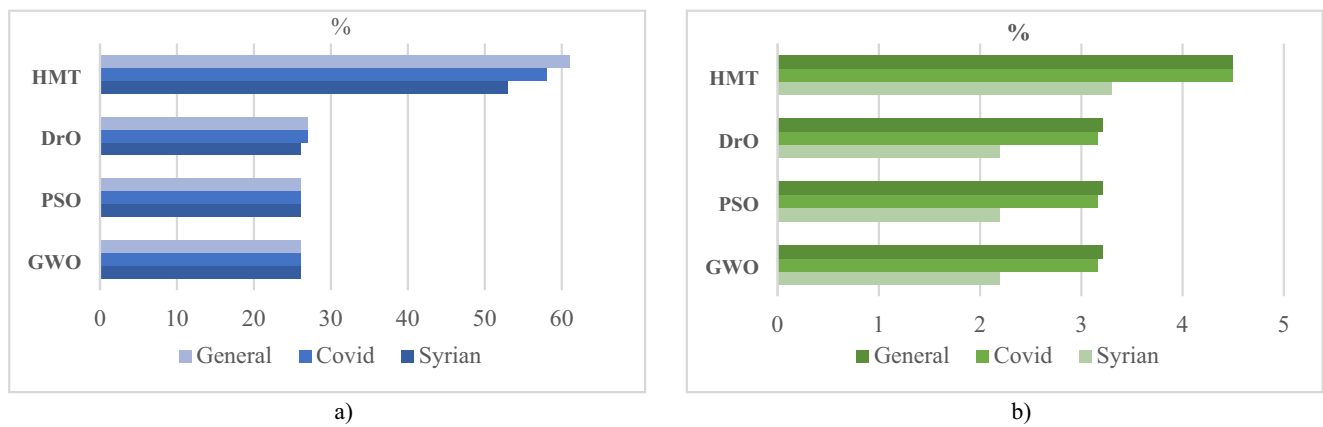


Fig. 12 CPU and memory loads of the used methods (a) CPU load (b) Memory load

in the True analysis. In fake news analyses, it was seen that words such as “image” and “watch” were often included in the best solution qualities.

Resource utilization in multi-thread applications is also an issue to be considered. Multi-thread and hybrid methods can increase resource usage. The CPU and memory loads of the HMT and single methods in the experiments are shown in Fig. 12a-b for comparison. Single methods generally consumed equal resources in all experiments, with a consumption of about 26% CPU and 2–3.5% memory (about 300–500 MB). On the other hand, the HMT consumed more CPU than the single-thread methods, with a CPU usage increasing of 53% for the Syrian dataset, 58% for the Covid data set, and 61% for the general-news data set. The memory load was approximately 1.3% higher than the single-threads methods.

8 Conclusion

Optimization-based approaches can offer flexible and efficient solutions for fake news detection. However, two main problems may arise with these approaches: a long runtime and inefficient use of search space. This study proposes the HMT model, which takes these two problems into account, with an additional multi-thread framework that is based on this model. The model allows the WrT threads to use different meta-heuristic algorithms that can run in parallel on the same search space. The most important feature of this model is that it uses a SvT that can monitor WrT threads at runtime. By examining the WrT patterns, the SvT thread can generate recommendation candidates that have the untested variable values in the optimization stage. Swarm-based GWO, PSO, and DrO with different mechanisms serve as meta-heuristic algorithms. Experiments show that HMT can achieve very successful and competitive results, especially in small and medium-sized fake news datasets. The proposed method also has the advantage of a higher interpretability provided by optimization-based approaches. For this, the average weights of the selected attributes, obtained from all experiments, were

considered. Thus, in the training phase, the most effective attributes that create patterns that exceed the similarity threshold can be seen. On the other hand, like all hybrid methods, the HMT could increase resource usage. In addition to the time saving and accuracy performance provided by the HMT, CPU resource consumption increased by 57.33% on average. In terms of memory, it consumed an average of 1.3% more resources in the experimental platform used compared to the single methods. Among single-thread methods, the GWO using the hunting mechanism is more successful. Among the well-known deep models and standard classification algorithms, the LSTM and SVM come to the front, respectively. The author’s next work will aim at fake news detection through an ensemble model that contains the HMT and a pre-trained deep network, to improve the results.

Appendix 1

In an operating system (OS), programs run inside processes. A process contains at least one thread (main thread). Threads can be created easily with the programming language used. Therefore, they are less costly than processes. There may be many processes on a computer at a given time, and they may want to use system resources simultaneously. Today’s multi-core CPU technologies (or hyper-threading technologies) enable parallel programming. Thus, more than one process or process thread can be run in parallel and the execution times can be shortened compared to sequential work. However, the number of threads that an OS will run is always more than the physical cores and because of this, an appropriate scheduling mechanism is required. Therefore, modern operating systems divide the time into parts called epochs and plan which processes or threads of the processes will run in each epoch (scheduling). Thus, the processes or process threads can use the processor in a time-division mode. The basic working principle of this mechanism is given in Fig. 13.

The scheduling, starting, and termination of process threads are called context-switching. A thread uses CPU resources (such

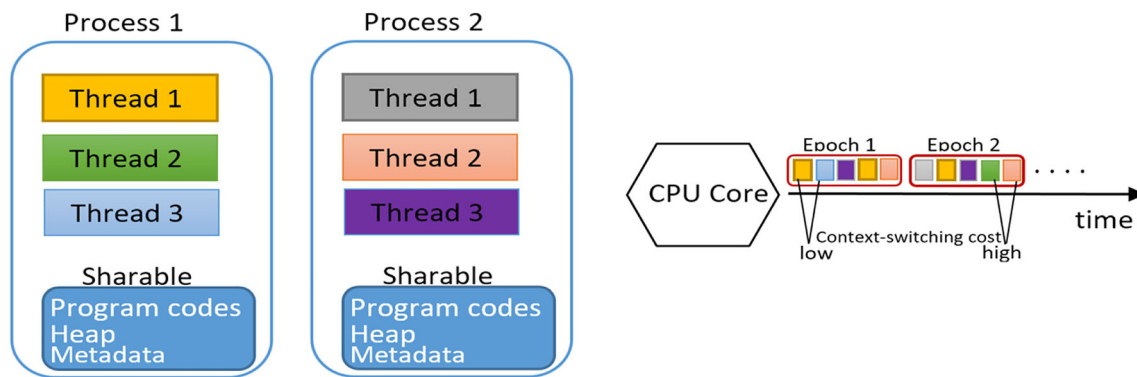


Fig. 13 The basic principle of multi-threaded execution

as registers) in its running time, and when the scheduled running time ends, all the data (variable values, the status of register flags, etc.) of that thread is saved until its next running time. The pre-saved data of the new thread is brought to the CPU. Naturally, context-switching is a costly mechanism. While this cost is higher between different processes, it is more suitable for in-process threads. This is because communication between processes is more complex, while threads within a process can share program codes, heap area in memory, and other meta-data. Since a hybrid parallel method in which metaheuristic algorithms run sequentially in separate processes with a single thread is too costly, the HMT is based on in-process multithreading. In the HMT, all metaheuristic algorithms are run in different threads within a process. A shared class (ShO) is used to avoid concurrency problems during data exchange and updating. In this shared object, there are variables (such as the best value) that can be updated only by one thread per unit of time.

References

1. Abedalla A, Al-Sadi A, Abdullah M (2019) A closer look at fake news detection: a deep learning perspective. In Proc of ICAAI:24–28. <https://doi.org/10.1145/3369114.3369149>
2. Ahmed H, Traore I, Saad S (2018) Detecting opinion spams and fake news using text classification. Journal of Security and Privacy 1(1):e9. <https://doi.org/10.1002/spy2.9>
3. Akyol S, Alatas B (2020) Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization. Physica A: Statistical Mechanics and its Applications 540:123094. <https://doi.org/10.1016/j.physa.2019.123094>
4. Bhattacharya P, Patel SB, Gupta R, Tanwar S, Rodrigues JJPC (2022) SaTYa: trusted bi-LSTM-based fake news classification scheme for smart community. IEEE Transactions On Computational Social Systems:1–10. <https://doi.org/10.1109/TCSS.2021.3131945>
5. Blum C, Puchinger J, Raidl G, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. Appl Soft Comput 11: 4135–4151. <https://doi.org/10.1016/j.asoc.2011.02.032>
6. Cao N, Ji S, Chiu DKW, Gong M (2022) A deceptive reviews detection model: separated training of multi-feature learning and classification. Expert Syst Appl 187:115977. <https://doi.org/10.1016/j.eswa.2021.115977>
7. Chen W, Zhang Y, Yeo CK, Lau CT, Lee BS (2018) Unsupervised rumor detection based on users' behaviors using neural networks, pattern recognition letters. Vol. 105:226–233. <https://doi.org/10.1016/j.patrec.2017.10.014>
8. Collins B, Hoang DT, Nguyen NT, Hwang D (2021) Trends in combating fake news on social media—a survey. J Inf Telecommun 5(2):247–266. <https://doi.org/10.1080/24751839.2020.1847379>
9. Fan Q, Han H, Wu S (2022) Credibility analysis of water environment complaint report based on deep cross domain network. 2021. Appl Intell. <https://doi.org/10.1007/s10489-021-02842-0>
10. Fernandez M, Bellogin A (2020) Recommender systems and misinformation: the problem or the solution? OHARS'20: Workshop on Online Misinformation- and Harm-aware Recommender Systems 2758:40–50
11. Gmys J, Pessoa TC, Melab N, Talbi E (2020) A comparative study of high-productivity high-performance programming languages for parallel metaheuristics. Swarm and Evolutionary Computation 57: 100720. <https://doi.org/10.1016/j.swevo.2020.100720>
12. Goldani, M. H., Momtazi, S., & Safabakhsh, R. (2021a). Detecting fake news with capsule neural networks. [arXiv:2002.01030](https://arxiv.org/abs/2002.01030)
13. Goldani MH, Safabakhsh R, Momtazi S (2021b) Convolutional neural network with margin loss for fake news detection. Inf Process Manag 58(1):102418. <https://doi.org/10.1016/j.ipm.2020.102418>
14. Irda YL, Irda SEF, Faizi R (2019) Automatic detection of fake news on online platforms: a survey. In Proc of ICSSD:1–4
15. Kennedy J, Eberhart R (1995) Particle swarm optimization. In Proc of IEEE:1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
16. Khanam Z, Alwasel NB, Sirafi H, Rashid M (2021) Fake news detection using machine learning approaches. In proc IOP 1099(1):012040. <https://doi.org/10.1088/1757-899X/1099/1/012040>
17. Kiziloluk S, Ozer AB (2017) Web pages classification with parliamentary optimization algorithm. Int J Softw Eng Knowl Eng 27(03):499–513. <https://doi.org/10.1142/S0218194017500188>
18. Klyuev V (2018) Fake news filtering: semantic approaches. In Proc of ICRITO:9–15. <https://doi.org/10.1109/ICRITO.2018.8748506>
19. Koirala A (2021) COVID-19 Fake News Dataset. *Mendeley Data*, V1, <https://doi.org/10.17632/zwfdmp5syg.1>
20. Kumari R, Ashok N, Ghosal T, Ekbal A (2021) Misinformation detection using multitask learning with mutual learning for novelty detection and emotion recognition. Inf Process Manag 58(5): 102631. <https://doi.org/10.1016/j.ipm.2021.102631>
21. Lalwani S, Sharma H, Satapathy SC, Deep K, Bansal JC (2019) A survey on parallel particle swarm optimization algorithms. Arab J Sci Eng 44:2899–2923. <https://doi.org/10.1007/s13369-018-03713-6>
22. Lotfi S, Mirzarezaee M, Hosseinzadeh M, Seydi V (2021) Detection of rumor conversations in twitter using graph convolutional networks. Appl Intell 51:4774–4787. <https://doi.org/10.1007/s10489-020-02036-0>

23. Mertoglu U (2020) A fake news detection model for Turkish language. PhD Thesis, Hacettepe University, Open Access System of Hacettepe University, Retrieved from <http://www.openaccess.hacettepe.edu.tr:8080/xmlui/handle/11655/23222?show=full>
24. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
25. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Applic* 27:1053–1073. <https://doi.org/10.1007/s00521-015-1920-1>
26. Nasir JA, Khan OS, Varlamis I (2021) Fake news detection: a hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights* 1(1):100007. <https://doi.org/10.1016/j.ijime.2020.100007>
27. Ozbay FA, Alatas B (2019) A novel approach for detection of fake news on social media using metaheuristic optimization algorithms. *Elektronika ir Elektrotechnika* 25(4):62–67. <https://doi.org/10.5755/j01.eie.25.4.23972>
28. Ozbay FA, Alatas B (2020) Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and its Applications* 540: 123174. <https://doi.org/10.1016/j.physa.2019.123174>
29. Ozbay FA, Alatas B (2021) Adaptive Salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media. *Multimed Tools Appl* 80:1–25. <https://doi.org/10.1007/s11042-021-11006-8>
30. Pavleska T, Skolkay A, Zankova B, Ribeiro NC, Bechmann A (2018) Performance analysis of fact-checking organizations and initiatives in Europe: a critical overview of online platforms fighting fake news. *Disinformation and Digital Media As A Challenge For Democracy*, vol 6. <https://doi.org/10.5281/zenodo.3677439>
31. Pennycook G, Rand DG (2019) Fighting misinformation on social media using crowdsourced judgments of news source quality. *Proc Natl Acad Sci* 116(7):2521–2526. <https://doi.org/10.1073/pnas.1806781116>
32. Połap D, Keşik K, Woźniak M, Damasevicius R (2021) Parallel technique for the metaheuristic algorithms using devoted local search and manipulating the solutions space. *Applied Science* 8(2):293. <https://doi.org/10.3390/app8020293>
33. Raj C, Meel P (2021) ConvNet frameworks for multi-modal fake news detection. *Appl Intell* 51:8132–8148. <https://doi.org/10.1007/s10489-021-02345-y>
34. Reis JC, Correia A, Murai FM, Veloso A, Benevenuto F (2019) Supervised learning for fake news detection. *IEEE Intell Syst* 34(2): 76–81. <https://doi.org/10.1109/MIS.2019.2899143>
35. Ren Y, Zhang J (2020) HGAT: hierarchical graph attention network for fake news detection. *Computer Science, Social and Information, Networks*. <https://doi.org/10.48550/arXiv.2002.04397>
36. Sahoo SR, Gupta BB (2021) Multiple features based approach for automatic fake news detection on social networks using deep learning. *Appl Soft Comput* 100(3):106983. <https://doi.org/10.1016/j.asoc.2020.106983>
37. Salem FK, Feel RA, Elbassuoni S, Jaber M, Farah M (2019) FA-KES: a fake news dataset around the Syrian war. In the Proc of AAAI 13(01):573–582
38. Samadi M, Mousavian M, Momtazi S (2021) Deep contextualized text representation and learning for fake news detection. *Inf Process Manag* 58(6):102723. <https://doi.org/10.1016/j.ipm.2021.102723>
39. Shabani S, Sokhn M (2018) Hybrid machine-crowd approach for fake news detection. In Proc of IEEE 1:299–306. <https://doi.org/10.1109/CIC.2018.00048>
40. Sharma K, Qian F, Jiang H, Ruchansky N, Zhang M, Liu Y (2019) Combating fake news: a survey on identification and mitigation techniques. *ACM Trans Intell Syst Technol (TIST)* 10(3):1–4 arXiv:1901.06437
41. Silva A, Han Y, Luo L, Karunasekera S, Leckie C (2021a) Propagation2Vec: embedding partial propagation networks for explainable fake news early detection. *Inf Process Manag* 58:102618. <https://doi.org/10.1016/j.ipm.2021.102618>
42. Silva CVM, Fontes RS, Junior MC (2021b) Intelligent fake news detection: a systematic mapping. *Journal of applied security research* 16(2):168–189. <https://doi.org/10.1080/19361610.2020.1761224>
43. Song C, Ning N, Zhang Y, Wu B (2021a) A multimodal fake news detection model based on crossmodal attention residual and multi-channel convolutional neural networks. *Inf Process Manag* 58(1): 102437. <https://doi.org/10.1016/j.ipm.2020.102437>
44. Song C, Shu K, Wu B (2021b) Temporally evolving graph neural network for fake news detection. *Inf Process Manag* 58(6):102712. <https://doi.org/10.1016/j.ipm.2021.102712>
45. Yildirim G (2022) A novel grid-based many-objective swarm intelligence approach for sentiment analysis in social media. *Neurocomputing* 503:173–188. <https://doi.org/10.1016/j.neucom.2022.06.092>
46. Yildirim G, Alatas B (2021) New adaptive intelligent grey wolf optimizer based multi-objective quantitative classification rules mining approaches. *J Ambient Intell Humaniz Comput* 12:9611–9635. <https://doi.org/10.1007/s12652-020-02701-9>
47. Zhou X, Zafarani R (2018) Fake news: a survey of research, detection methods, and opportunities. ArXiv. <https://doi.org/10.48550/arXiv.1812.00315>
48. Zhou X, Zafarani R (2019) Network-based fake news detection: a pattern-driven approach. *ACM SIGKDD Explor Newsletter* 21(2): 48–60 arXiv:1906.04210

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Gungor Yildirim was born in Elazig, Turkey. He received the B.Sc. degree from Firat University, Turkey, in electrical and electronic engineering. He received the M.Sc and the PH.D degrees from computer engineering of Firat University, in 2012, and 2017, respectively. He worked as a project and control engineer in private sectors and a public institution (DSI). He served as the head of the electric program at Tunceli MYO for three years. In 2016, he received the Award of Appreciation in DSI. His research

interests include wireless systems, wireless sensor networks, and multi-many optimizations.