A multi-swarm optimisation approach for spam detection in online social networks

R. Krithiga* and E. Ilavarasan

Department of Computer Science and Engineering,
Pondicherry Engineering College,
Tamil Nadu, Puducherry, India
Email: kriithiga@gmail.com
Email: eilavarasan@pec.edu
*Corresponding author

Abstract: Online Social Networks (OSNs) play a crucial role in communication systems for rapid message broadcasting and information sharing. Facebook is one of the popular OSNs that has the highest number of active users as per the statistical reports. However, it witnesses challenges due to the presence of spammers, whose actions may make the environment unfavourable for users. The spammers hold social accounts that are particularly created for personal benefit. As the number of users on Facebook increases, the number of illegitimate accounts proportionally rises, which leads to distress in the OSN environment. Several methods have been proposed in the literature to address the spam profile detection problem; however, they become obsolete as the spammers evolve. Hence, in this paper, a novel Multi-Swarm-Whale Optimisation Algorithm (MS-WOA) is proposed for feature selection to detect spam profiles on Facebook. Further IP-address-based features tailored for Facebook are also proposed. The performance of the proposed MS-WOA is compared with the recently developed methods and outperforms them in terms of accuracy and robustness.

Keywords: multi-swarm approach; whale optimisation algorithm; spam detection; social networks; information and communication systems.

Reference to this paper should be made as follows: Krithiga, R. and Ilavarasan, E. (2021) ‘A multi-swarm optimisation approach for spam detection in online social networks’, *Int. J. Vehicle Information and Communication Systems*, Vol. 6, No. 1, pp.64–87.

Biographical notes: R. Krithiga currently works as an Assistant Professor in the Department of Computer Applications at Perunthalaivar Kamarajar Arts College, Puducherry. She is also a Research Scholar at Pondicherry Engineering College. Her area of interest includes evolutionary algorithms, data mining, machine learning and intelligent systems.

E. Ilavarasan is a Professor in the Department of Computer Science and Engineering at Pondicherry Engineering College, Puducherry. He has experience of more than 25 years in the teaching field. His area of specialisation is web service computing.

Copyright © 2021 Inderscience Enterprises Ltd.

1 Introduction

With the advancements in the internet and mobile technologies, communication has become sophisticated, revolutionary and futuristic (Fire and Elovici, 2014). Social networks play a crucial role in communication systems for rapid message broadcasting and information sharing. These are actively used to share information within a network of users. It is being used by individuals, organisations, companies, etc., to share intended contents to the target audience. Facebook is one of the popular Online Social Network (OSN) platforms and has the highest number of users across the world, according to the statistics reported by Statista (2020). This popularity has also fascinated the intruders to take advantage of the network for personal gain. The intruders are social network accounts termed as spam profiles that share unsolicited messages or links with ill-intentions through malicious or harmful links, contents and cloaking (Wang et al., 2013). And there are also singleton spammers and community spamming groups that are prevalent in OSN (Kumar et al., 2018). In this paper, we attempt to identify spammers on Facebook. The present scheme adopted by Facebook requires manual reporting and this has called for further investigation and research on the problem. The spamming in OSN could be broadly classified into two classes: (1) Spam content detection and (2) Spam node detection (Elakkiya and Selvakumar, 2020; Adewole et al., 2019). By detecting the spam contents, the texts or posts that have the spamming material could only be flagged. Nonetheless, detecting spam messages alone would not help in curbing spammers. Detecting the spam accounts that initiate spamming is to be identified as it would ultimately stop the spread of spam. One of the critical challenges confronted by the researchers to develop spammer detector systems is the evolving nature of spammers. A system devised during a particular time spell could not be successfully applied for its intended purpose as the smart spammers evade the system (Miller et al., 2014). The existing researches on Facebook lack a rich set of features for effective detection. This necessitates the need to devise spammer identifier systems that are robust, scalable, online, and computationally less expensive. The spammers are defined by a sequence of their characteristics or features. The feature selection is a prominent pre-processing step in data mining to determine the optimal subset of features that could contribute to classifying an instance from a given set of N features (Cai et al., 2018). No single feature is meticulously discriminative between spam and legitimate profiles. Every feature is likely to fail in certain scenarios. Conversely, when used collectively even the weaker features tend to exhibit powerful discrimination ability (Liu et al., 2018). The filter methods rank individual features based on some statistical evaluation. The wrapper method on the other hand focuses on selecting a subset of features (Xu and Wang, 2011). In this paper, seven Multi-Swarm variants of the Whale Optimisation Algorithm (MS-WOA) are proposed for wrapper-based feature selection to identify spam accounts on Facebook. The design and network structure of social networks differ from each other. Hence, a system developed for a particular type of social network may not necessarily operate well on others. Further, IP-address-based features are employed to effectively identify the spammers concealed in the network.

The remainder of the paper is organised as follows: Section 2 provides a preliminary foundation about the proposed work and the state-of-the-art of spam detection in the Facebook. Section 3 explores the proposed variants of MS-WOA and Section 4 discloses the experimental results and findings. Section 5 concludes the paper with summaries on future work and scope.

2 Background information and literature survey

2.1 The native WOA

The Whale Optimisation Algorithm (WOA) is one of the recently developed optimisation algorithms and has been successfully utilised in the pre-processing steps of data mining and feature selection in particular. WOA has been proposed by Seyedali Mirjalili (Mirjalili and Lewis, 2016) based on the prey hunting behaviour of whales. One of the advantages of WOA is that the probability of falling onto the local optima is comparatively lesser than the other standard population-based algorithms such as GA and PSO (Mohammed et al., 2019). The whales exhibit few peculiar manoeuvres for foraging and this behaviour has been modelled to suit the optimisation problems. The whales exhibit mechanisms such as (a) encircling prey, (b) bubble net attacking and (c) random search. The exploration is achieved by the random search and exploitation by the other two strategies. These movements exhibited by the whales are mathematically formulated as given below:

a) *The encircling strategy*:The whales encircle the prey for foraging and thus the position of the whales for the next iteration is updated using equation (2).

$$\vec{D} = \vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t) \tag{1}$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \tag{2}$$

where $\vec{X}(t)$ denotes the position of whales during the current iteration, $\overrightarrow{X^*}(t)$ represents the best whale of the current iteration, *A* and *C* are calculated using the following equations (3), (4) and (5).

$$\vec{A} = 2.\vec{a}.\vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r} \tag{4}$$

$$a = 2 - t\frac{2}{t_{\text{Max}}} \tag{5}$$

where a takes the value between 2 and 0 in the decreasing fashion, *r* is a random vector in the range [0, 1], t_{Max} is the threshold set for the iteration count.

b) *Bubble net attacking strategy*:

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t) \tag{6}$$

$$\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{7}$$

$$\vec{X}(t+1) = \begin{array}{ll} \overrightarrow{X^*}(t).\vec{A}.\vec{D} & \text{if} \quad p < 0.5 \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t) & \text{if} \quad p < 0.5 \end{array} \tag{8}$$

where $\overrightarrow{D'}$ is the distance between the *i*-th individual and the best whale, *b* describes the shape of the spiral and is preset in the algorithm, *l* is a random number generated in the range [–1, 1].

This strategy mimics the spiral upward movement of the whales and moves to a new location-based on equation (6). The choice between encircling mechanism and bubble-net attacking is decided based on the probability p as given in equation (8).

c) *Prey (Random) search*:

$$\vec{D} = \vec{C}.\overrightarrow{X_{rand}}(t) - \vec{X}(t) \tag{9}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}}(t) - \vec{A}.\vec{D} \tag{10}$$

where $\overrightarrow{X_{rand}}$ is a random individual chosen from the population.

Depending on the value of $|A|$, the algorithm chooses between the encircling mechanism and the random search. The algorithm Native WOA starts with generating a random initial population. The fitness of the whales is then evaluated and the individuals with the highest accuracy become the best whales of that iteration. For a certain number of iterations set as the stopping criterion, the whales repeat refreshing their position following a particular strategy based on the values of p and $|A|$ as in Figure 1. Finally, the best whale is output as the optimal solution.

Figure 1 Algorithm of native – WOA

```
Algorithm: Native WOA
Initialise WOA parameters
Randomly generate initial population of whales $W_{ij}\ (j = 1, 2, ..., m)$
Evaluate the fitness of whales
Sort the solutions based on the accuracy and choose the best solution
best_whale*= best solution
$t$=0
while t<max_iterations do
    for 3 each solution represented by the whale do
        if $p$<0.5 then
            if $|A|$<1 then
                position updating using encircling strategy
            Else
                position updating using random search
        Else
            position updating using bubble net attacking strategy
    end for
    update best_whale*
    $t = t + 1$
end while
output best_whale*
```

WOA has been widely applied to solve many real-world problems such as image segmentation (El Aziz et al., 2017), wind speed forecasting (Wang et al., 2017), flow shop scheduling problem (Abdel-Basset et al., 2018), sentiment analysis (Tubishat et al., 2019), feature selection (Hussien et al., 2019a), forecasting gold price fluctuations (Alameer et al., 2019), etc., yet suffers from several limitations such as

- Poor search space exploration (Mohammed et al., 2019)
- Weak management of exploration and exploitation (Luo and Shi, 2019)
- Lack of techniques to jump out of local optima (Luo and Shi, 2019)
- Increased computational time (Zhanga et al., 2019)
- Longer time for convergence (Qiao et al., 2019)
- The encircling strategy would further forward the algorithm to get trapped in local optima (Bozorgi and Yazdani, 2019).

2.2 Literature survey

Social networks pose several problems that open up new research dimensions such as link prediction, trust measurement, friend recommendation, rumour source detection, user profiling and fake news detection (Bliss et al., 2014; Gong et al., 2020; Cai and Xu, 2019; Shelke and Attar, 2019; Ikeda et al., 2013; Aldwairi and Alwahedi, 2018). Spam account detection is one such problem that has been researched on all kinds of OSN platforms (Barushka and Hajek, 2018; Prieto et al., 2013; Fu et al., 2018; Yusof and Sadoon, 2017). Gao et al. (2012) collected Facebook's wall posts and analysed for similarity in the URL destination or content of the post. The study concluded that most of the spamming originated from compromised accounts rather than fake accounts. Most of the spam detection works on Facebook have predominantly considered features based on wall posts. This framework would be serviceable if the intention is to only find the spam posts. Ahmed and Abulaish (2013) proposed a generic set of features to be applied to Facebook and Twitter based on the visible interactions through these platforms. Naïve Bayes classifier delivered a good performance for the Facebook data set and J48 for the combined data set. Further, the discrimination ability of each feature was evaluated using information gain.

In Chawla (2014), the Facebook pages were analysed using the *n*-gram model using the SVM classifier and concluded that the unigram model yields a comparatively best performance. However, the *n*-gram model may fail on multi-lingual data sets. Although networks such as Twitter and Instagram offer follow and following functionalities, only Facebook (FB) offers a distinctive feature of placing friend requests and making friends on the OSN. Many existing FB users accept a friend request from unknown people. Once this request is accepted, it implicitly suggests that the users are mutually following each other. Hence, these community interaction-based features were employed in Bhat and Abulaish (2013) and evaluated the performance using four classifiers such as Decision Tree, J48, *K*-NN and Naïve Bayes. The investigation demonstrated the effectiveness of community-based features. This study was later extended and evaluated for ensemble classifiers (Bhat et al., 2014). In the experiment conducted, the bagging ensemble approach based on J48 produced a TP rate and a low FP rate with 0.966 and 0.082, respectively. A Markov clustering algorithm was used in Setiawan et al. (2016) to reduce the effect of spammers on FB. As a case study, the method was tested on the Javanese FB user data set and the work focused on features such as active friends, number of likes and URL. The study inferred that spam profiles exchange page likes to gain reputation, share URL with the same domain address, and mutually tagging on a wall post. Based on the publicly available profile features, a real-time spam detection system was proposed in Rathore et al. (2017). The method used features based on profile and content and achieved an accuracy of 0.972 using a random forest classifier. Recently, the optimisation algorithms have been widely applied in the feature selection phase to the effective discovery of spammers. Aswani et al. (2018) employed an enhanced Firefly Algorithm (FA) with a *K*-means algorithm to detect spammers in Twitter marketing. The Chaotic Optimisation Algorithm was further used to tune the absorption co-efficient and attractiveness co-efficient. In Sohrabi and Karimi (2018), a PSO-based feature selection method was performed on the FB data set using the SVM classifier for the spam detection problem. Al-Zoubi et al. (2018) employed Whale Optimisation Algorithm (WOA) to detect spammers in the multi-lingual Twitter dataset. Hence, in this work, it is attempted to improve this performance by enhancing the ability of WOA. The WOA has several improved versions devised for specific problems. Abdel-Basset et al. (2019) improved WOA was proposed to solve the 0–1 knapsack problem. The method was improved in terms of the introduction of a local search strategy and Levy flight to search for regions. Sun et al. (2018) enhanced WOA primarily to address the disadvantage of the control parameter $|A|$. It adopted a cosine function in the control parameter to effectively balance exploration and exploitation to solve global optimisation problems. Since ideal solution is not known in the beginning phase of WOA, the search may lead the solution beyond the optimal value and this was addressed in Khadanga et al. (2020) by introducing correction factors. While several methods have been proposed in the literature for spammer detection, the spammers smartly evade the detection mechanism by impersonating legitimate users.

3 The proposed multi-swarm-based whale optimisation algorithm (MS-WOA)

In this paper, seven variants of binary-WOA are proposed for feature selection to detect spam profiles on Facebook. The multi-swarm is an emerging paradigm that has been proven to produce exciting results at a greater convergence rate (Qiu, 2019). Figure 2 lists the variants of WOA proposed based on a simple multi-swarm strategy (variant –1), neighbouring swarm-based (variants 2–4), and shuffled mechanism (variants 5–7). These multiple swarms independently deal with the objective of the problem and also through information sharing. This information sharing among the swarms would facilitate faster convergence at global optimum. The proposed WOA variants also treat the overlapping locations chosen by the whales. When a whale occupies the same position as that of another whale of a different swarm, then the position is re-initialised to diversify the search and broaden the searching space. The proposed methods start with an equal-sized sub-population. Even if one sub-swarm gets stuck in the local optima, the other sub-swarms can continue the search process or jump out of the local optima by relocating its position based on the global best position. Each whale is represented as a binary sequence of ‘0’s and ‘1’s that indicates the presence or absence of a particular feature.

The accuracy of the classifier is considered as the fitness value and is given as in equation (11). Hence, the greater the accuracy value better the model is. More details on TP, TN, FP and FN are provided in Section 4.

$$\text{Fitness}\left(W_{ij}\right) = \text{ Accuracy of}\left(W_{ij}\right) = \frac{TP+TN}{TP+FP+TN+FN} \tag{11}$$

where w_{ij} represents a $whale_i$ of $swarm_j$ and is denoted as a sequence of feature values. Since the conventional WOA is not suitable for classification tasks, the continuous search space is transformed into a binary space using the sigmoid function Hussien et al. (2019a) as in equation (13).

$$S\left(\overrightarrow{W_{ij}}\right) = \frac{1}{1+e^{-\left(W_{ij}\right)}} \tag{12}$$

$$W_{ij} = \begin{cases} 1, \ \& rand \geq 0\,\text{Sigmoid}\left(\overrightarrow{W_{ij}}\right) \\ \qquad 0, \ \& \,\text{otherwise} \end{cases} \tag{13}$$

Figure 2 Proposed variants of binary WOA

2. *Proposed variants of WOA*	
Multi-Swarm based on the interactiveness among the swarms	
Variant – 1	Multi-swarm WOA with no interaction
Multi-swarm WOA based on the interactiveness among the swarms	
Variant – 2	Multi-swarm WOA where the whales update the position with respect to the best whale of the succeeding swarm
Variant – 3	Multi-swarm WOA where the whales update the position with respect to the best whale of the preceding swarm
Variant – 4	Multi-swarm WOA where the whales update the position with respect to the best whale among the entire swarm
Multi-swarm WOA with a shuffled mechanism	
Variant – 5	Shuffled Multi-swarm WOA, where whales are shuffled among the swarms after generations following the position updating mechanism given in variant – 2.
Variant – 6	Shuffled Multi-swarm WOA, where whales are shuffled among the swarms after generations following the position updating mechanism given in variant – 3.
Variant – 7	Shuffled Multi-swarm WOA, where whales are shuffled among the swarms after generations following the position updating mechanism given in variant – 4.

Variant – 1: This variant consists of multiple populations (swarms) of whales where each swarm behaves in much the same way as the native WOA. As in Figure 3, for *t* iterations, the algorithm operates independently and the best position of a whale from a swarm is considered to be the optimal solution. This variant is very much similar to running multiple instances of native WOA in parallel and choosing the best value obtained by an instance of WOA. This method does not share the information with other swarms and continue to autonomously function to achieve the goal. Hence, there is no interaction among the swarms. The native WOA guides all the solutions towards the best position. However, in this variant, the whales of each sub-swarm are guided by the best solution of the respective swarm. The disadvantage of this method is that if a sub-swarm suffer from local optima, it will get entrapped and cannot have any means to escape from the local optima. However, the other sub-swarms proceed to evolve.

Figure 3 Algorithm of WOA – variant – 1

```
Algorithm – WOA variant – 1
Initialise WOA parameters
For$_1$ each swarm $S_i\left(i=1,2,...,k\right)$ do
    Randomly generate initial population of whales $W_{ij}\left(j=1,2,...,m\right)$
    Evaluate the fitness of whales
end for$_1$
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale$_i$*= the best solution of swarm $S_i$
$t$=0
while$_1$ $t$<max_iterations do
    for$_2$ each swarm $S_i$ do
        for$_3$ each solution represented by the whale do
            update the position of whales based on leader_whale$_i$
        end for$_3$
    end for$_2$
    Transfer the positions onto a binary space
    Resolve for overlapping positions of whales
    Update leader_whale$_i$
    $t=t+1$
end while$_1$
g_leader=max(leader_whale$_i$)
Output g_leader
```

Variant – 2: This variant based on the interactiveness among the sub-swarms possesses two classes of whales called "member whales" and "leader whales". The algorithm is shown in Figure 4. The best whales of each of the sub-swarm are termed as *leader whales* whereas the rest of the whales in the sub-swarm are characterised as *member whales*. During iteration, the member whales may also become leader whales for a particular swarm. The leader whales are the local best positions in each of the sub-swarms. In this method, the leader whale of swarm$_i$ updates the position based on the leader whale of swarm$_{i+1}$, whereas the member whales of sub-swarms update the position based on the corresponding leader whales. Thus the relocation of leader whales in this fashion greatly diversifies the search space.

Figure 4 Algorithm of WOA – variant – 2

```
Algorithm: WOA variant – 2
Initialise WOA parameters
for_1 each swarm $S_i\left(i=1,2,...,k\right)$ do
    Randomly generate initial population of whales $W_{ij}\left(j=1,2,...,m\right)$
    Evaluate the fitness of whales
end for_1
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale_i*= the best solution of swarm $S_i$
$t$=0
while_1 t<max_iterations do
    for_2 each swarm $S_i$ do
            for_3 each solution represented by the whale do
                if_1 the whale== leader_whale_i
                        Update the position of whales based on leader_whale_{i+1}
                else_1
                        Update the position of whales based on leader_whale_i
               end if_1
            end for_3
    end for_2
    Transfer the positions onto a binary space
    Resolve for overlapping positions of whales
    Update leader_whale_i
    $t=t+1$
end while_1
g_leader =max(leader_whale_i)
Output g_leader
```

The algorithm starts with generating a random binary population of whales called individual solutions. The fitness values of each of these solutions are evaluated and the leader whales of each sub-swarm are found. After the intra-swarms update their position based on the position of the leader, the inter-swarm position update takes place. This is achieved by allowing the leader whale of swarm$_i$ to update its position based on the position of the leader whale of swarm$_{i+1}$. Suggestively the leader whale of a swarm follows the leader whale of the next immediately succeeding swarm. The subsequent leader whales update their positions in this manner. The entire process is repeated for a definite number of iterations. Once the stopping criterion is met, the leader whales of the entire swarm are compared against each other for the best fit. The best whale is termed as the global leader and corresponds to the feature sequence of the spam detection problem.

Variant – 3: The leader whales in this variation update their positions based on the position of leader whales of the preceding swarm. Hence, a leader whale of swarm$_i$ achieves migration with respect to the leader whale of swarm$_{i-1}$. The rest of the procedures are similar to the steps followed for variant – 2 and are displayed in Figure 5.

Figure 5 Algorithm of WOA – variant – 3

```
Algorithm: WOA variant – 3
Initialise WOA parameters
for$_1$ each swarm $S_i\left(i=1,2,...,k\right)$ do
    Randomly generate initial population of whales $W_{ij}\left(j=1,2,...,m\right)$
    Evaluate the fitness of whales
end for$_1$
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale$_i$*= the best solution of swarm $S_i$
$t$=0
while$_1$ $t$<max_iterations do
    for$_2$ each swarm $S_i$ do
        for$_3$ each solution represented by the whale do
            if$_1$ the whale== leader_whale$_i$
                Update the position of whales based on leader_whale$_{i-1}$
            else$_1$
                Update the position of whales based on leader_whale$_i$
            end if$_1$
        end for$_3$
    end for$_2$
    Transfer the positions onto a binary space
    Resolve for overlapping positions of whales
    Update leader_whale$_i$
    $t=t+1$
end while$_1$
g_leader =max(leader_whale$_i$)
Output g_leader
```

Variant – 4: This version of WOA starts with initialising algorithm-specific WOA parameters. The whales in the sub-swarms are randomly generated with binary vectors and evaluated for fitness. As given in Figure 6, before entering the main loop of the method, the algorithm first chooses the leader-whale and global leader. The leader whales update their positions based on the global leader and the member whales travel towards the leader whale of the respective swarm. In every iteration, the leader whales and the global leader are updated.

Figure 6 Algorithm of WOA – variant – 4

```
Algorithm: WOA variant – 4
Initialise WOA parameters
for$_1$ each swarm $S_i\left(i=1,2,...,k\right)$ do
    Randomly generate initial population of whales $W_{ij}\left(j=1,2,...,m\right)$
    Evaluate the fitness of whales
end for$_1$
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale$_i$*= the best solution of swarm $S_i$
g_leader =max(leader_whale$_i$*)
$t$=0
while$_1$ $t$<max_iterations do
    for$_2$ each swarm $S_i$ do
            for$_3$ each solution represented by the whale do
                if$_1$ the whale== leader_whale$_{i–i}$
                        Update the position of whales based on gbest
                else$_1$
                        Update the position of whales based on leader_whale$_i$
              end if$_1$
            end for$_3$
    end for$_2$
    Transfer the positions onto a binary space
    Resolve for overlapping positions of whales
    Update leader_whale$_i$ and g_leader
    $t=t+1$
end while$_1$
Output g_leader
```

The variants from 5 through 7 are based on the shuffling mechanism. After the lapse of a certain number of iterations set as a threshold, the algorithm retains the leader whales and shuffles the member whales among the sub-swarms. Though the earlier versions brought in diversification by implicitly shuffling the leader whales, these versions intensify and strengthen the diversification process. These variations provide interactions as well as bring in diversification among the swarms. The previous versions are embedded with shuffling mechanism and are presented as variant – 5 (Refer Figure 7), variant – 6 (Refer Figure 8), and variant – 7 (Refer Figure 9) by enhancing the variants 2, 3 and 4.

Figure 7 Algorithm of WOA – variant – 5

```
Algorithm: WOA variant – 5
Initialise WOA parameters
for1 each swarm $S_i\left(i=1,2,...,k\right)$ do
    Randomly generate initial population of whales $W_{ij}\left(j=1,2,...,m\right)$
    Evaluate the fitness of whales
end for1
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale$_i$*= the best solution of swarm $S_i$
shuffle=0
while1 shuffle<max_shuffle do
    $t$=0
    while2 $t$<max_iterations do
        for2 each swarm $S_i$ do
            for3 each solution represented by the whale do
                if1 the whale== leader_whale$_i$
                    Update the position of whales based on leader_whale$_{i+1}$
                else1
                    Update the position of whales based on leader_whale$_i$
                end if1
            end for3
        end for2
        Transfer the positions onto a binary space
        Resolve for overlapping positions of whales
        Update leader_whale$_i$
        $t=t+1$
    end while2
    randomly associate the whale$_{ij}$ to a swarm$_k$ (where $i\neq$k)
    shuffle=shuffle+1
end while1
g_leader =max(leader_whale$_i$)
Output g_leader
```

Figure 8 Algorithm of WOA – variant – 6

```
Algorithm: WOA variant – 6
Initialise WOA parameters
for_1 each swarm $S_i \left(i = 1, 2, ..., k\right)$ do
    Randomly generate initial population of whales $W_{ij} \left(j = 1, 2, ..., m\right)$
    Evaluate the fitness of whales
end for_1
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale$_i$*= the best solution of swarm $S_i$
shuffle=0
while_1 shuffle<max_shuffle do
    $t$=0
    while_2 $t$<max_iterations do
        for_2 each swarm $S_i$ do
            for_3 each solution represented by the whale do
                if_1 the whale== leader_whale$_i$
                    Update the position of whales based on
                    leader_whale$_{i-1}$
                else_1
                    Update the position of whales based on
                    leader_whale$_i$
                end if_1
            end for_3
        for_2 end
        Transfer the positions onto a binary space
        Resolve for overlapping positions of whales
        Update leader_whale$_i$ and g_leader
        $t = t + 1$
    end while_2
    randomly associate the whale$_{ij}$ to a swarm$_k$ (where $i$≠k)
    shuffle=shuffle+1
end while_1
g_leader =max(leader_whale$_i$)
Output g_leader
```

Figure 9 Algorithm of WOA – variant – 7

Algorithm: WOA variant – 7

```
Initialise WOA parameters
for_1 each swarm $S_i\ (i = 1, 2, ..., k)$ do
    Randomly generate initial population of whales $W_{ij}\ (j = 1, 2, ..., m)$
    Evaluate the fitness of whales
end for_1
Sort the solutions of each swarm based on the accuracy
Elect the best solutions of each swarm based on the highest accuracy
leader_whale_i*= the best solution of swarm $S_i$
g_leader =max(leader_whale_i*)
shuffle=0
while_1 shuffle<max_shuffle do
    $t$=0
    while_2 $t$<max_iterations do
        for_2 each swarm $S_i$ do
            for_3 each solution represented by the whale do
                if_1 the whale== leader_whale_i
                    Update the position of whales based on g_leader
                else_1
                    Update the position of whales based on
                    leader_whale_i
                end if_1
            end for_3
        end for_2
        Transfer the positions onto a binary space
        Resolve for overlapping positions of whales
        Update leader_whale_i
        $t = t + 1$
    end while_2
    randomly associate the whale_ij to a swarm_k (where $i \neq k$)
    shuffle=shuffle+1
end while_1
g_leader =max(leader_whale_i)
Output g_leader
```

4 Experiment analysis and results

The performance of the proposed seven variants of WOA is analysed in terms of accuracy, precision, recall and *F*-measure. These metrics are determined from the confusion matrix constructed as in Table 1. The precision is calculated as per equation (14) and is defined as the number of Facebook spam profiles correctly classified by the system to the total number of Facebook profiles that have been labeled as spam by the system. The recall, as given in equation (15) is defined as the ratio of the number of Facebook spam profiles correctly labeled by the system to the total number of actual spam profiles. The *F*-measure is a harmonic mean of precision and recall and is calculated as per equation (16).

Table 1 Confusion matrix for spammer detection

	Predicted: Spam	*Predicted: Non-spam*
Actual: Spam	True Positive (TP)	False Negative (FN)
Actual: Non-spam	False Positive (FP)	True Negative (TN)

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

where

TP – Number of Facebook profiles correctly recognised as spam

TN – Number of Facebook profiles misrecognised as non-spam

FP – Number of Facebook profiles incorrectly recognised as spam

FN – Number of Facebook profiles misrecognised as non-spam

4.1 Information on dataset

As Facebook's policy does not let individuals share the data, a public dataset to study and investigate the spam detection problem is not available. Using Facebook's API, a crawler was utilised to fetch and construct the dataset. The dataset collected and constructed resulted in 2634 legitimate profiles and 867 spam profiles. A total of 24 features are used in the problem. Nineteen features based on profile and content are adapted from Rathore et al. (2017) and five novel features are also proposed as a part of the contribution to this work. As the method developed runs on the server-side, simple and effective features are utilised to manage the server load. Performing too many calculations and the inclusion of complex features would put the server down. The study in Qian et al. (2010) states that the spammers' IP addresses stay active only for 24 hours. Spamming has been considered as an automated process and so IP-address-based features that were proposed in our earlier work (Krithiga and Ilavarasan, 2020) have been tailored for Facebook and presented in Table 2.

Table 2 IP address-based features

Feature	*Description*
Feature_{20}	No. of the distinct IP address in a week time
Feature_{21}	No. of the distinct IP address in the last 24 hours
Feature_{22}	No. of posts deleted
Feature_{23}	No. of IP Addresses used for posts
Feature_{24}	No. of IP Addresses used for comments

4.2 Parameter settings

As SVM is widely applied in the spam profile detection problem (Sohrabi and Karimi, 2018) and Adaboost delivered an outstanding performance in our previous research with spam detection on Twitter (Krithiga and Ilavarasan, 2020), these two classifiers are considered for comparison. All the methods are coded in Python and executed in a Windows 10, 64-bit machine with 8 GB RAM, intel core–i7 configuration. The default environment of Scikit learn was set for SVM and AdaBoost. The number of whales is set to 6, and iterations to 15. The results are averaged over 20 runs. The training and testing samples were chosen in the ratio 70:30. Table 3 lists the techniques used in the work for performance evaluation.

Table 3 Methods considered for comparison

Methods	*Algorithms*
Technique – 1	Native WOA
Technique – 2	WOA Variant – 1
Technique – 3	WOA Variant – 2
Technique – 4	WOA Variant – 3
Technique – 5	WOA Variant – 4
Technique – 6	WOA Variant – 5
Technique – 7	WOA Variant – 6
Technique – 8	WOA Variant – 7
Technique – 9	The method proposed in Rathore et al. (2017)
Technique – 10	The method proposed in (Sohrabi and Karimi (2018)

Case 1: The first experiment is conducted without performing feature selection using the classifiers and the results are presented in Figure 10. The performance of Adaboost yields better results than SVM with an accuracy of 88.44%.

Case 2: This mode of the experiment allows feature selection to be performed before the classification task. The results of algorithms with the SVM classifier is displayed in Figure 11 and that of AdaBoost in Figure 12.

Figure 10 Comparison with baseline classifiers

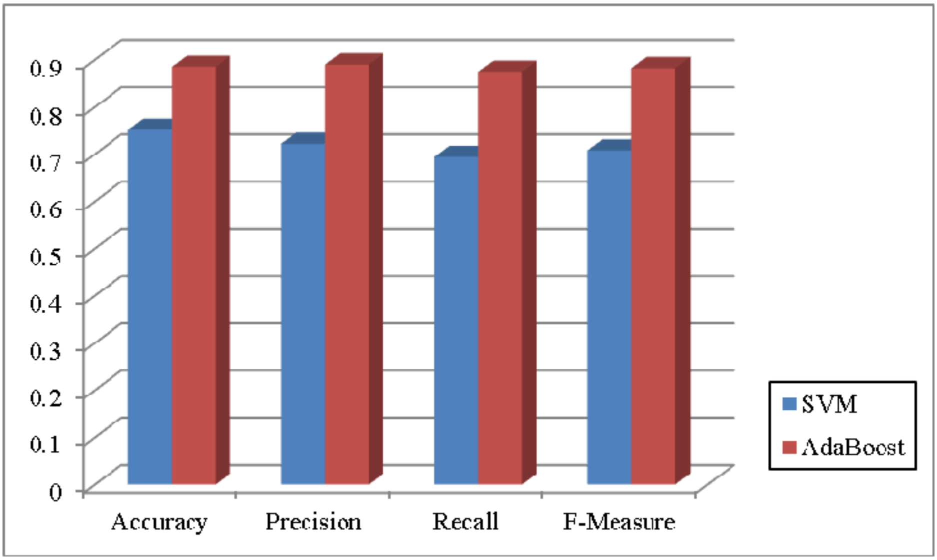


Figure 11 Performance of algorithms using SVM

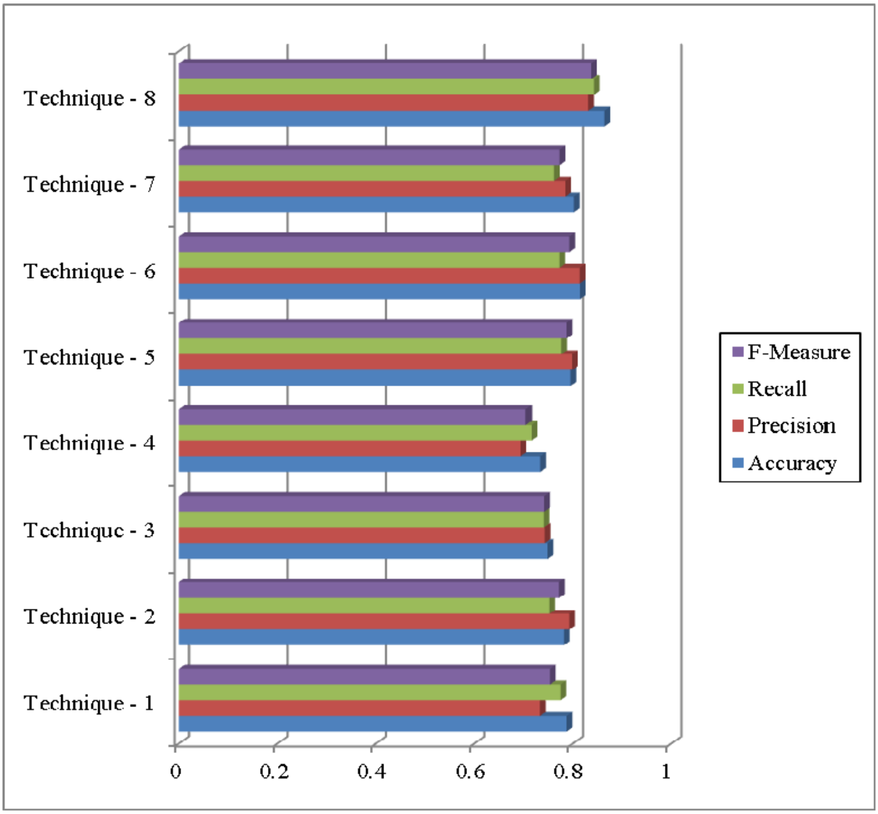
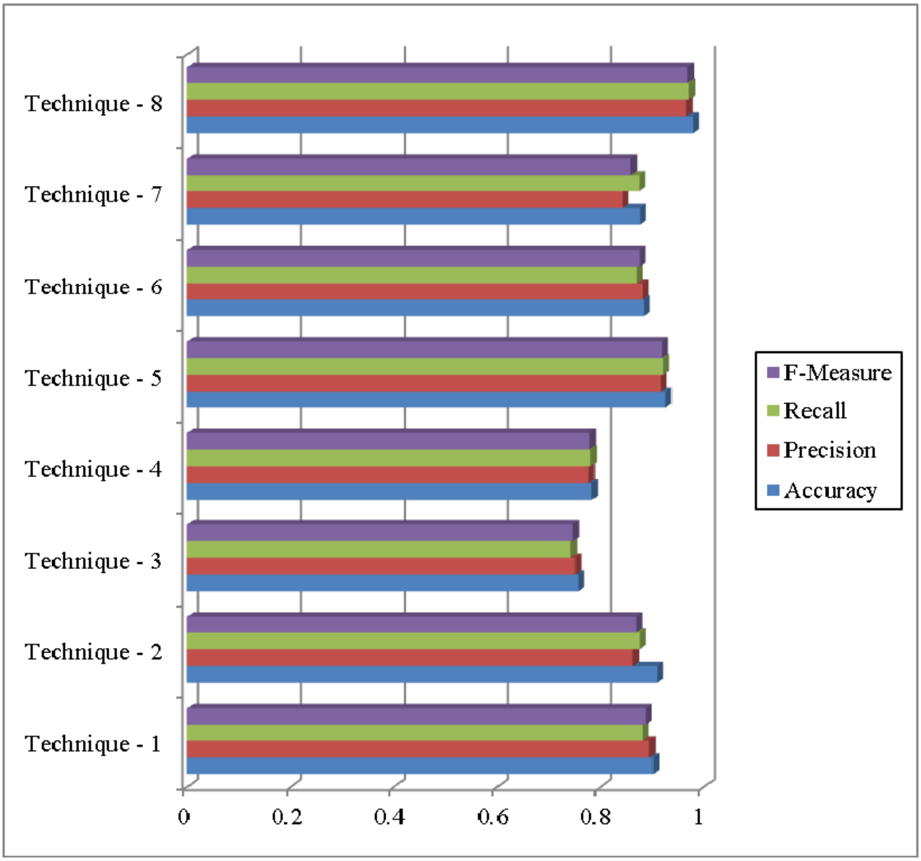


Figure 12 Performance of algorithms using AdaBoost

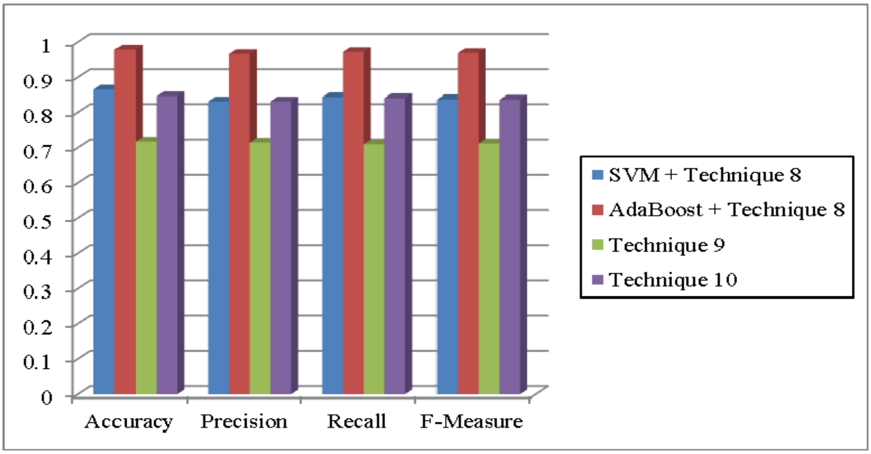


The performance of feature selection algorithms with SVM classifier can be ranked as follows: Technique 8 > Technique 6 > Technique 7 > Techniques > Technique 5 > Technique 3 > Technique 4 > Technique 1 > Technique 2. The comparison with AdaBoost is graded as follows: Technique 8 > Technique 5 > Technique 1 > Technique 6 > Technique 2 > Technique 7 > Technique 4 > Technique 3.

In summary, the shuffled mechanism of WOA performs better than the other versions of WOA as the randomisations are seeded aiding better diversification. The native WOA performs better than the simple multi swarm and neighboring swarm variants using the Adaboost classifier. Another observation from the results is that all the variants improves in performance upon employing Adaboost classifier. Therefore, the choice of classifier significantly increases performance.

Case 3: The best models in the literature are compared along with the proposed variants, and results are presented in Figure 13, which shows that the methods proposed in the earlier works are not effective with the change in time. That is, when the method is applied to a dataset extracted at a later period, fails to produce convincing results. This also implies that the spammers do evolve and evade the system.

Figure 13 Performance comparison with the existing methods



Case 4: The features selected by the top models are presented in Figure 14 with AdaBoost and WOA variant – 7 selecting the smaller subset of features with a count of 14. In order to obtain further insights on the features selected by the algorithm, the frequency count was also calculated. A counter was assigned to every feature and during iteration, the corresponding counters for features marked as '1' were incremented. In this fashion, Figure 15 shows the frequency of features for 15 iterations. It shows the frequency count of all the selected 14 features by the AdaBoost + WOA variant – 7. It is to be mentioned that the entire proposed IP-address-based features have been included in the subset and chosen by the algorithm. This proves the effectiveness and robustness of the proposed IP-address-based feature set.

Figure 14 No. of features selected by the best top models

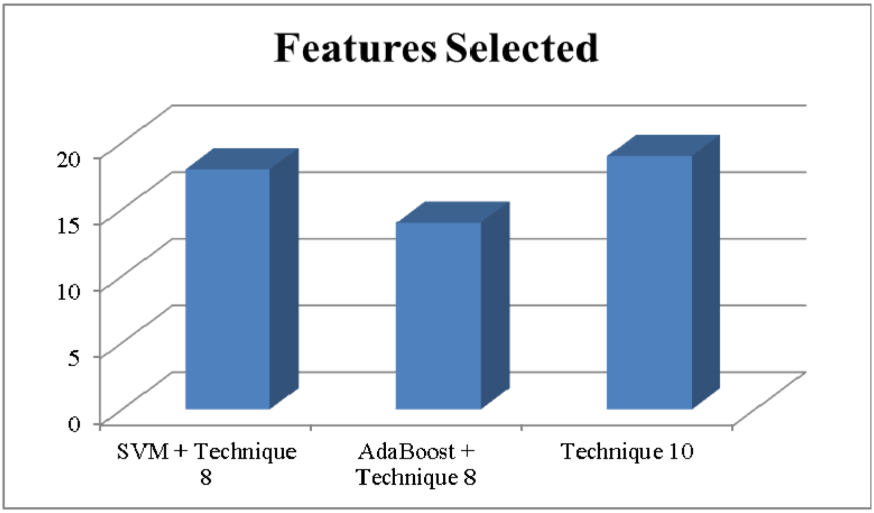
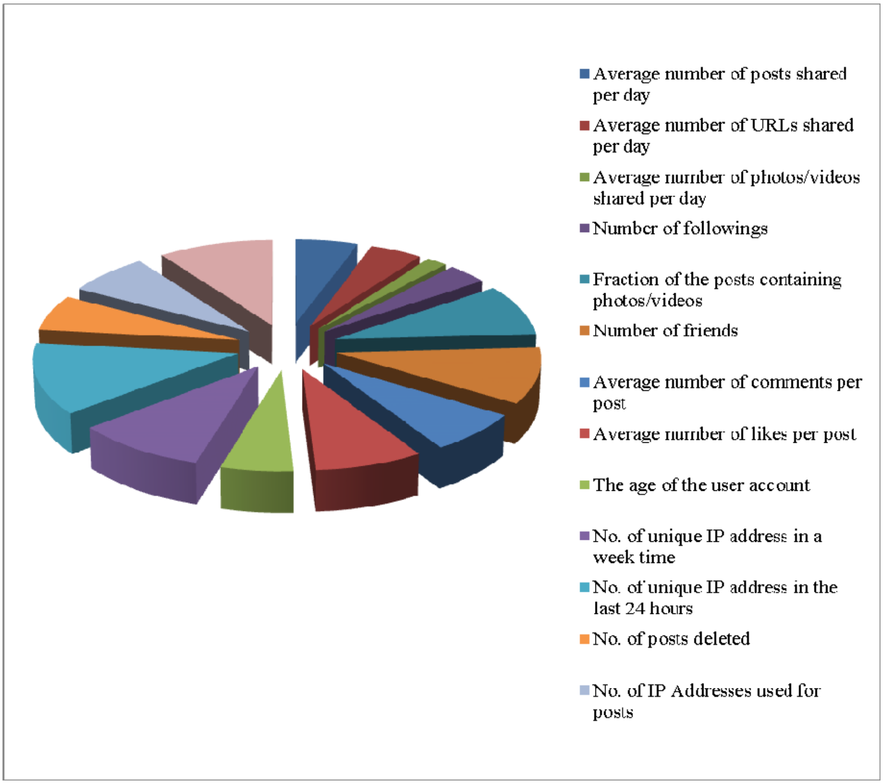


Figure 15 Frequency count of feature subset selected by the best variant



Finally, the cumulative ranking of the proposed variants is provided in Table 4. The WOA – variant 7 tops the ranking; the native WOA performs better than variants 1, 2 and 3. As the leader of the swarm traverses towards the global best leader, the mechanism fails to address the possibility of the global best leader whale falling into the trap of local optima. However, it is the shuffling mechanism that keeps the searching process diversified and also helps to escape from local optima.

Table 4 Cumulative ranking of WOA – variants based on this performance

WOA – Version	*Rank*
Native WOA	5
WOA Variant - 1	6
WOA Variant - 2	7
WOA Variant - 3	8
WOA Variant - 4	2
WOA Variant - 5	3
WOA Variant - 6	4
WOA Variant - 7	1

5 Conclusion and future work

In this paper, we have proposed seven variants of WOA to intensify the exploration and facilitate convergence at a faster rate. The proposed approach customised for feature selection was applied to detect spam accounts on Facebook. Further, IP-address-based features were utilised to tackle the evolving spammers and keep the system robust. The presence of these features made it possible to get rid of re-training and thereby making it also appropriate for online detection. The variant – 1 of WOA yields results similar to that of the native WOA. The shuffled variant with position updating based on global leader using AdaBoost outperformed the other versions in terms of accuracy, *F*-measure and smaller subset of features. Hence, it is concluded that the variant – 7 of WOA using AdaBoost possessed a strong discriminating ability in distinguishing Facebook spam profiles with an accuracy of 97.82%.

We aim to extend this work by developing an integrated system to be operational across social networks. We also intend to extract Facebook user profile data of three different periods with 6 months interval and evaluate the applicability of the proposed method.

References

Abdel-Basset, M., El-Shahat, D. and Sangaiah, A.K. (2019) 'A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem', *International Journal of Machine Learning and Cybernetics*, Vol. 10, pp.495–514.

Abdel-Basset, M., Manogaran, G., El-Shahat, D. and Mirjalili, S. (2018) 'A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem', *Future Generation Computer Systems*, Vol. 85, pp.129–145.

Adewole, K.S., Anuar, N.B., Kamsin, A. and Sangaiah, A.K. (2019) 'SMSAD: a framework for spam message and spam account detection', *Multimedia Tools and Applications*, Vol. 78, pp.3925–3960.

Ahmed, F. and Abulaish, M. (2013) 'A generic statistical approach for spam detection in online social networks', *Computer Communications*, Vol. 36, Nos. 10/11, pp.1120–1129.

Alameer, Z., Elaziz, M.A., Ewees, A.A., Ye, H. and Jianhua, Z. (2019) 'Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm', *Resources Policy*, Vol. 61, pp.250–260.

Aldwairi, M. and Alwahedi, A. (2018) 'Detecting fake news in social media networks', *Proceedings of the 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN'18)*, Vol. 141, pp.215–222.

Al-Zoubi, A.M., Faris, H., Alqatawna, J. and Hassonah, M.A. (2018) 'Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts', *Knowledge-Based Systems*, Vol. 153, No. 1, pp.91–104.

Aswani, R., Kar, A.K. and Ilavarasan, P.V. (2018) 'Detection of spammers in twitter marketing: a hybrid approach using social media analytics and bio inspired computing', *Information System Frontiers*, Vol. 20, pp.515–530. Doi: 10.1007/s10796-017-9805-8.

Barushka, A. and Hajek, P. (2018) 'Spam filtering in social networks using regularized deep neural networks with ensemble learning', in Iliadis, L., Maglogiannis, I. and Plagianakos, V. (Eds): *Proceedings of the International Conference on Artificial Intelligence Applications and Innovations*, Springer, Vol. 519, pp.38–49.

Bhat, S.Y. and Abulaish, M. (2013) 'Community-based features for identifyingspammers in online social networks', *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp.100–107.

Bhat, S.Y., Abulaish, M. and Mirza, A.A. (2014) 'Spammer classification using ensemble methods over structural social network features', *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'14)*, 11–14 August, Warsaw, Poland, pp.454–458.

Bliss, C.A., Frank, M.R., Danforth, C.M. and Dodds, P.S. (2014) 'An evolutionary algorithm approach to link prediction in dynamic social networks', *Journal of Computational Science*, Vol. 5, No. 5, pp.750–764.

Bozorgi, S.M. and Yazdani, S. (2019) 'IWOA: an improved whale optimization algorithm for optimization problems', *Journal of Computational Design and Engineering*, Vol. 6, pp.243–259.

Cai, C. and Xu, H. (2019) 'A topic sentiment based method for friend recommendation in online social networks via matrix factorization', *Journal of Visual Communication and Image Representation*, Vol. 65. Doi: 10.1016/j.jvcir.2019.102657.

Cai, J., Luo, J., Wang, S. and Yang, S. (2018) 'Feature selection in machine learning: a new perspective', *Neurocomputing*, Vol. 300, pp.70–79.

Chawla, H. (2014) 'Facebook page spam detection using support vector machines based on n-gram model', *International Journal of Computer Science Issues*, Vol. 11, No. 5, pp.161–163.

El Aziz, M.A., Ewees, A.A. and Hassanien, A.E. (2017) 'Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation', *Expert Systems with Applications*, Vol. 83, pp.242–256.

Elakkiya, E. and Selvakumar, S. (2020) 'GAMEFEST: genetic algorithmic multi evaluation measure based feature selection technique for social network spam detection', *Multimedia Tools and Applications*, Vol. 79, pp.7193–7225.

Fire, M. and Elovici, Y. (2014) 'Online social networks: threats and solutions', *IEEE Communication Surveys and Tutorials*, Vol. 16, No. 4, pp.2019–2033.

Fu, Q., Feng, B., Guo, D. and Li, Q. (2018) 'Combating the evolving spammers in online social networks', *Computers and Security,* Vol. 72, pp.60–73.

Gao, H., Chen, Y., Lee, K., Palsetia, D. and Choudhary, A. (2012) 'Towards online spam filtering in social networks', *Proceedings of the 19th Annual Network and Distributed System Security Symposium*, pp.1–17.

Gong, Z., Wang, H., Guo, W., Gong, Z. and Wei, G. (2020) 'Measuring trust in social networks based on linear uncertainty theory', *Information Sciences*, Vol. 508, pp.154–172.

Hussien, A.G., Hassanien, A.E., Houssein, E.H., Bhattacharyya, S., Amin, M. (2019a) 'S-shaped binary whale optimization algorithm for feature selection', in Bhattacharyya, S., Mukherjee, A., Bhaumik, H., Das, S. and Yoshida, K. (Eds): *Recent Trends in Signal and Image Processing. Advances in Intelligent Systems and Computing*, Springer, Singapore, Vol. 727, pp.79–87. Doi: 10.1007/978-981-10-8863-6_9.

Hussien, G., Hassanien, A.E., Houssein, E.H., Amin, M. and Azar, A.T. (2019b) 'New binary whale optimization algorithm for discrete optimization problems', *Engineering Optimization*, Vol. 52, No. 6, pp.945–959.

Ikeda, K., Hattori, G., Ono, C., Asoh, H. and Higashino, T. (2013) 'Twitter user profiling based on text and community mining for market analysis', *Knowledge-Based Systems*, Vol. 51, pp.35–47.

Khadanga, R.K., Kumar, A. and Panda, S. (2020) 'A novel modified whale optimization algorithm for load frequency controller design of a two-area power system composing of PV grid and thermal generator', *Neural Computing and Applications*, Vol. 32, pp.8205–8216.

Krithiga, R. and Ilavarasan, E. (2020) 'A novel hybrid algorithm to classify spam profiles in twitter', *Webology*, Vol. 17, No. 1, pp.260–279.

Kumar, D., Shaalan, Y., Zhang, X. and Chan, J. (2018) 'Identifying singleton spammers via spammer group detection', *Advances in Knowledge Discovery and Data Mining* (PAKDD'18), pp.656–667.

Liu, X-Y., Liang, Y., Wang, S., Yang, Z-Y. and Ye, H-S. (2018) 'A hybrid genetic algorithm with wrapper-embedded approaches for feature selection', *IEEE Access*, Vol. 6, pp.22863–22874.

Luo, J. and Shi, B. (2019) 'A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems', *Applied Intelligence*, Vol. 49, pp.1982–2000.

Miller, Z., Dickinson, B., Deitrick, W., Hua, W. and Wang, H.A. (2014) 'Twitter spammer detection using data stream clustering', *Information Sciences*, Vol. 260, pp.64–73.

Mirjalili, S. and Lewis, A. (2016) 'The whale optimization algorithm', *Advances in Engineering Software*, Vol. 95, pp.51–67.

Mohammed, H.M., Umar, S.U. and Rashid, T.A. (2019) 'A systematic and meta-analysis survey of whale optimization algorithm', *Computational Intelligence and Neuroscience*, pp.1–25.

Prieto, V.M., Alvarez, M. and Cacheda, F. (2013) 'Detecting linkedin spammers and its spam nets', *International Journal of Advanced Computer Science and Applications*, Vol. 4, No. 9, pp.189–199.

Qian, Z., Mao, Z., Xie, Y. and Yu, F. (2010) 'On network-level clusters for spam detection', *NDSS Symposium*, pp.1–17.

Qiao, W., Huang, K., Azimi, M. and Han, S. (2019) 'A novel hybrid prediction model for hourly gas consumption in supply side based on improved whale optimization algorithm and relevance vector', *IEEE Access*, Vol. 7, pp.88218–88230.

Qiu, C. (2019) 'A novel multi-swarm particle swarm optimization for feature selection', *Genetic Programming and Evolvable Machines*, Vol. 20, pp.503–529.

Rathore, S., Loia, V. and Park, J.H. (2017) 'SpamSpotter: an efficient spammer detection framework based on intelligent decision support system on facebook', *Applied Soft Computing Journal*, pp.920–932. Doi: 10.1016/j.asoc.2017.09.032.

Setiawan, E.I., Susanto, C.P., Santoso, J., Sumpeno, S. and Purnomo, M.H. (2016) 'Preliminary study of spam profile detection for social media using Markov clustering: case study on Javanese people', *Proceedings of the International Computer Science and Engineering Conference (ICSEC)*, Chiang Mai, pp.1–4. Doi: 10.1109/ICSEC.2016.7859942.

Shelke, S. and Attar, V. (2019) 'Source detection of rumor in social network –A review', *Online Social Networks and Media*, Vol. 9, pp.30–42.

Sohrabi, M.K. and Karimi, F. (2018) 'A feature selection approach to detect spam in the facebook social network', *Arabian Journal for Science and Engineering*, Vol. 43, pp.949–958.

Statista (2020) *Most popular social networks worldwide as of October 2020, ranked by number of active users*. Available online at: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Sun, Y., Wang, X., Chen, Y. and Liu, Z. (2018) 'A modified whale optimization algorithm for large-scale global optimization problems, *Expert Systems With Applications*, Vol. 114, pp.563–577. Doi: 10.1016/j.eswa.2018.08.027.

Tubishat, M., Abushariah, M.A.M., Idris, N. and Aljarah, I. (2019) 'Improved whale optimization algorithm for feature selection in Arabic sentiment analysis', *Applied Intelligence*, Vol. 49, pp.1688–1707.

Wang, J., Du, P., Niu, T. and Yang, W. (2017) 'A novel hybrid system based on a new proposed algorithm – multi-objective whale optimization algorithm for wind speed forecasting', *Applied Energy*, Vol. 208, pp.344–360.

Wang, W., Lee, X-D., Hu, A-L. and Geng, G-G. (2013) 'Co-training based semi-supervised web spam detection', *Proceedings of the 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Shenyang, China, pp.789–793.

Xu, Y. and Wang, H. (2011) 'A new feature selection method based on support vector machines for text categorisation', *International Journal of Data Analysis Techniques and Strategies*, Vol. 3, No. 1, pp.1–20.

Yusof, Y. and Sadoon, O.H. (2017) 'Detecting video spammers in youtube social media', in Zulikha, J. and Zakaria, N.H. (Eds): *Proceedings of the 6th International Conference of Computing and Informatics*, pp.228–234.

Zhanga, H., Tanga, L., Yangb, C. and Lan, S. (2019) 'Locating electric vehicle charging stations with service capacity using the improved whale optimization algorithm', *Advanced Engineering Informatics*, Vol. 41.