Received July 28, 2021, accepted August 1, 2021, date of publication August 3, 2021, date of current version August 12, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3102277*

# Metaheuristic Ant Lion and Moth Flame Optimization-Based Novel Approach for Automatic Detection of Hate Speech in Online Social Networks

**CEM BAYDOGAN**[1] **AND BILAL ALATAS**[2]

[1]Department of Software Engineering, Faculty of Technology, Firat University, 23119 Elâziğ, Turkey
[2]Department of Software Engineering, Faculty of Engineering, Firat University, 23119 Elâziğ, Turkey

Corresponding author: Bilal Alatas (balatas@firat.edu.tr)

**ABSTRACT** In the online social networks, blogs, microblogs, social bookmarking services and sharing sites, and various web forum pages; the sharing of knowledge, opinions, ideas, etc. are spreading very quickly. This situation brings very dangerous problems in social networks. One of these problems is hate speech detection (HSD) problem which is covering issues such as insults, swearing, humiliation, discrimination, exclusion, detest, abhor, blast, damn, and intolerance. These can be reactions to a person, a group, an organization, an order, or an event. Although few machine learning methods have been used in the literature to solve this important problem in online social media, the performance of the HSD models in terms of many metrics needs to be increased. In this study, an automatic HSD system based on metaheuristic methodology was proposed for better results in this new and important problem. In the proposed optimization approach, Ant Lion Optimization (ALO) algorithm and Moth Flame Optimization (MFO) algorithm were designed for the HSD problem. This is the first attempt to use optimization algorithms as solution search strategies for automatic HSD. An efficient representation scheme and flexible fitness function were designed for this purpose. Many metrics can easily be embedded into the designed fitness function in order to be simultaneously optimized. Firstly, the basic natural language processing (NLP) steps were carried out. Feature extraction was performed using Bag of Words (BoW), Term Frequency (TF), and document vector (Word2Vec). Then, the performances of the proposed novel approaches were analyzed in detail on the three different real-world data. The obtained results were also checked against eight popular supervised machine learning algorithms, Social Spider Optimization (SSO) algorithm, and state-of-the-art Tunicate Swarm Algorithm (TSA). Considering the evaluation criteria for three sets of experiments, it was observed that the accuracy, sensitivity, precision, and f-score results of the ALO and MFO algorithms were superior to machine learning methods. As a result of the experimental studies, the highest accuracy value was 92.1% for ALO, while this value was 90.7% for MFO. Other numerical values obtained in the study were given in the experiments and results section with tables and graphics in detail. Due to the promising results of the proposed approaches, they are anticipated to be used in the solution of many social media and networking problems.

**INDEX TERMS** Hate speech detection, metaheuristic optimization, natural language processing, social network analysis, text mining.

## I. INTRODUCTION

Hate speech is a new concept in social network terminology. Hence, hate speech does not have a universal definition. Hate speech is a wide field of study, including racist hatred, xenophobia, anti-Semitism, aggressive nationalism and immigrant nationalism, discrimination, sexual orientation, asylum, and refugee. The person(s) who disseminate, provoke, promote or legitimize content that includes one of these issues is defined as guilty. In fact, hate speech is widely accepted as a problem confused with freedom of expression. It would be more accurate to evaluate hate speech as spiritual violence.

Especially with the increase in the use of online social networks, it has increased the publishing of disturbing content

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

targeting personal, national, ethnic, and religious groups, etc. These shared contents are called hate speech, which is considered a digital crime. It is a new subject of study discovered by computer and data scientists. There are many social media and networking problems in the literature such as sentiment analysis [1], fake news detection [2], rumor detection [3], cyberbullying detection [4], customer satisfaction detection [5], link prediction [6], etc. The most important feature that distinguishes hate speech from other problems is that people who post on social networks think that they use their freedom of expression deliberately or unintentionally. However, content shared with the aim of destroying fundamental rights and freedoms cannot be considered within the scope of freedom of expression due to the fact that it is an abuse of the right and that expression containing hate and other rights and freedoms conflict. This is because attacks on the autonomy and self of the person or group targeted with hate speech will be paved and the discriminatory attitude of the discourse owners by not struggling with this discourse will result in the normalization of the discourse in the long run.

Automatic HSD is necessary to protect the areas, people, or groups that are targeted by hate speech. Therefore, automatic HSD is also important to protect against the abuse of hate speech. Moreover, automatic HSD is an important system to prevent acts that incite violence, discrimination, and hatred based on nationality, race, and religion.

HSD is a new social media analysis problem and limited works are consisting of supervised machine learning algorithms. However, the obtained values of different metrics according to the problem are not at desired level. The design, development, and implementation of new and efficient methods are important task under the philosophy of continuous improvement and always searching for better in many problems such as HSD.

Optimization techniques are used to solve problems by determining the best values for a goal function considering a set of available parameters satisfying the constraint eauality and inequality functions [7]. Metaheuristic techniques are more popular over classical optimization techniques due to simplicity and robustness [8]. Metaheuristic techniques are general purposed methodologies and they can easily and efficiently be used in different fields [9], [10].

To the best of our knowledge, optimization has never been used for solving the HSD problem in the literature. Therefore, this study is a reference resource that seeks to solve the HSD problem with optimization perspective. In addition, with this study, a new direction was developed for the solution of the HSD problem, which is one of the social network problems. The fundamental contributions of this study to the literature and science can be summarized as follows;

(1) Metaheuristic optimization algorithms that can solve many complex real-time world problems are used for the first time to automatically solve the HSD problem.
(2) Together with this study, a new solution search method for social network problems is proposed.
(3) A new form of representation is proposed for the solution of optimization algorithms on textual documents.
(4) A flexible fitness function is designed in this study. Many metrics can easily be embedded in the designed fitness function in order to be simultaneously optimized.
(5) With the proposed representation scheme and fitness function, many social network problems seem to be easily and efficiently be solved by the improved metaheuristic algorithms.
(6) A new problem field for the optimization algorithms is also introduced by this study.
(7) The results obtained are also compared with SSO and state-of-the-art TSA.

The remaining of the study is organized as follows: In the second section, the literature about the topic is searched. The most important reason for the shortness of literature is that the number of studies on this topic with metaheuristic optimization approach is very low. The characteristics and working features of the metaheuristic methods used in the study are mentioned in the third section. The algorithms are chosen based on their success in real-world problems. In the fourth section, used datasets and NLP methods with data pre-processing steps are explained. The methods used for feature extraction are explained in this section. Then, the machine learning methods used in the study are listed under the subtitle. Finally, in this section, how the intelligent optimization algorithms are modeled for the HSD problem is stated. Algorithm parameters are shown in this subtitle. Numerical information about the experimental studies conducted is given in the fifth section. The information obtained is shown in detail in tables and graphics. The success and ranking of the algorithms are performed in this section. In the last section of the study, conclusions and recommendations are presented. The strengths and weaknesses of the study are emphasized. Inferences are also made regarding future studies.

## II. RELATED STUDIES

Recently, studies on online social networks related to the HSD problem have been increasing in the literature. The HSD, which has become a popular study area, is an issue that requires automatic detection to prevent all people from being harmed. Santosh and Aravind used multi-language data in their HSD study on mixed social media text [11]. They obtained an accuracy of 70.7% on average using the word n-gram method. Rohan *et al.* used the transformed word embedding model in their HSD study on Twitter. They obtained an accuracy of 92% in this study [12]. Zeerak conducted a study on Twitter about racism and sexism [13]. In his study on 130K tweets, he achieved high performance in the recall, precision, and f-score by the token uni-gram method.

Aymé conducted HSD studies on 6 different datasets to show that HSD is not as easy as think. Researchers using deep learning algorithms obtained an f-score between 23% and 96% in the study [14]. Juan *et al.* used 6K data in their

HSD study on Twitter [15]. The LSTM+MLP neural network methodology proposed by the team achieved an accuracy value of 83%. Shervin and Marcos used the support vector machine classifier on the data consisting of 15K tweets in their HSD study. Using the n-gram method, they obtained an accuracy of 78% [16]. In his Ph.D. thesis, Shanita sought solutions to the HSD problem by using NLP techniques [17]. 91% accuracy value was achieved using the convolutional neural network. Michele *et al.* used supervised machine learning algorithms such as support vector machines, neural networks, and logistic regression. The obtained f-score on the three different datasets was 80% [18].

David *et al.* used seven different datasets for HSD on Twitter. They achieved more than 90% accuracy in their study using support vector machines and deep neural networks [19]. William and Julia conducted the HSD study on Yahoo using uni-gram, bi-gram, and tri-gram techniques. As a result of their work, they obtained an accuracy of 94% and recall at 60% [20]. Zeerak and Dirk used the n-gram method for the HSD problem [21]. Using 136K data in total, they achieved an f-score of 74% and a precision value of 72% on Twitter. Valentino *et al.* conducted the HSD for the Italian language. The team used the support vector machines and worked on a total of 1234K Italian tweets. In addition, they obtained f-score values above 80% [22]. Nina and Els used support vector machine methods in the HSD problem [23]. They reached an f-score of approximately 79% using 10K tweets.

One of the review articles for HSD was prepared by Macavaney *et al.* [24]. They explained common definitions used for hate speech in his work. The authors defined the datasets used for HSD to cover the year in which the study was conducted. They listed the approaches (word-based approaches, machine learning approaches, etc.) used for HSD. Finally, they presented a reference paper for future studies by comparing the experimental results of previous studies.

HSD system supporting multi languages was proposed by Aluru *et al.* [25]. They used deep learning techniques in a total of 9 languages including Arabic, English, German, Indonesian, Italian, Polish, Portuguese, Spanish, and French. The datasets, they used in their work, were obtained from social networks such as Twitter, Facebook, and Stormfront. They used 4 different models: MUSE+CNN-GRU, Translation+BERT, LASER+LR, and mBERT. They achieved the highest performance values in Arabic, English, Indonesian, Italian, and Spanish datasets with their proposed mBERT model.

Sigurbergsson *et al.* aimed to HSD study on Danish and English datasets [26]. They used Logistic Regression and Bi-LSTM methods on the data collected from Twitter, Facebook, and Reddit social networks. The highest results were obtained with the Bi-LSTM method. In the Danish dataset, they achieved the highest recall value as 70%. In the English dataset, the precision value was 77%. Furthermore, an f-score value of 72% was obtained in the English dataset.

García-Díaz *et al.* conducted an HSD study on misogyny against women on Spanish tweets [27]. They divided into 3 sub-dataset as violence against women, harassment against women, and misogyny against women. They used Random-Forest, Sequential Minimal Optimization, and LSVM machine learning classifier methods. Among all these methods, they achieved the highest 85.2% accuracy with the Sequential Minimal Optimization algorithm. Word embeddings and NLP techniques were used, which raised awareness of violence against women in social networks.

Mossie and Wang proposed an HSD study on the Amharic language in Ethiopia [28]. They created a dataset as Amharic texts with hate speech data collected from Facebook pages. They completed their studies using GBT, RF, RNN, and RNN+LSTM classifiers. They used TF-IDF, Word2Vec, unigram, and bi-gram feature models in their experiments. They achieved 92% accuracy by using the RNN architecture and Word2Vec embedding feature.

Djuric *et al.* conducted an HSD study on the data collected from the Yahoo Finance website for 6 months [29]. They carried out their studies by proposing the BoW+TF, BoW+TF-IDF, and paragraph2vec models. In the study, they conducted using 5-fold cross-validation. They obtained an accuracy of 74% with BoW models. With the paragraph2vec model, they achieved the highest accuracy value on this dataset.

Ghosh-Roy *et al.* proposed an HSD model using the Perspective API [30]. Ghosh-Roy *et al.* conducting HSD studies using English, German, and Hindi languages used the deep multi-layer perceptrons method. They obtained the highest 91% accuracy in the English dataset, the highest 82% in the German dataset, and the highest 75% accuracy in the Hindi dataset, respectively.

The remarkable studies on HSD in recent years with the gradual advancement, as well as the current level of the state-of-the-art and the differenet metrics are summarized in Table 1. When the literature is examined, different HSD problems have been solved with very different methods. However, to the best of our knowledge there is not intelligent optimization based model for HSD problems. We adapted intelligent ALO and MFO methods as solution search strategy for HSD problem in order to increase the success of the HSD model in terms of different metrics. It is aimed to bring a new breath to the literature with the proposed methods. In addition, a different perspective is aimed to be presented to the solution of other social network and media problems such as the HSD problem.

## III. OVERVIEW OF METAHEURISTIC OPTIMIZATION ALGORITHMS

Metaheuristic optimization algorithms are a sub-branch of artificial intelligence and are inspired by the intelligent behavior of living creatures, ant colonies, insects, bees, and various fish species. There are many metaheuristic optimization algorithms in the literature such as swarm-based, physics-based, hybrid-based, chemistry-based, sports-based,

**TABLE 1.** Literature research on HSD studies.

| Ref. No | Year | Classifiers/Algorithms /Frameworks/Tools/Models | Dataset | Obtained Metrics (max. values) | Used Feature Methods | Definition |
|---|---|---|---|---|---|---|
| **[31]** | 2017 | Support Vector Machine, Naive Bayes, k-Nearest Neighbors | Kenya Hate Speech Dataset | Accuracy: 0.7027<br>Precision: 0.70<br>Recall: 0.70<br>F-score: 0.70 | Uni-gram, bi-gram, and tri-gram features and TF-IDF | TF-IDF + n-grams based approach for HSD |
| **[32]** | 2017 | Twitter API Tool | Italian Twitter Corpus | Annotation category accuracy ratio: 0.54 | Cohen's kappa | Creating an HS dataset with Italian twitter corpus |
| **[33]** | 2017 | Support Vector Machine | German Hate Speech Twitter Corpus | Macro-averaged F-score: 0.65 | Uni-gram, bi-gram, and tri-gram features | Examining the effect of implicit on automatic HSD |
| **[34]** | 2017 | FastText Framework, CNNs and LSTMs. | Annotated tweets | Precision: 0.93<br>Recall: 0.93<br>F-score: 0.93 | TF-IDF and BoW | Deep learning approach for HSD |
| **[35]** | 2017 | Logistic Regression, Naive Bayes, Decision Trees, Random Forests, and LinearSVMs. | Hate speech and offensive language dataset | Precision: 0.91<br>Recall: 0.90<br>F-score: 0.90 | Lexical–based methods ( Uni-gram, bi-gram, and tri-gram features, TF-IDF) | Machine learning approach for HSD (Lexical-based) |
| **[36]** | 2018 | Logistic Regression, Multilayer Perceptron, LSTM, and CNN+GRU | Wikipedia and Twitter Hate Speech Dataset | F-score: 0.86 | n-gram features on both the character-level and word-level | HSD with character-level features and word-level features |
| **[37]** | 2018 | Random Forest, Support Vector Machine, and J48 | Three Different Twitter Hate Speech Dataset | Precision: 0.883<br>Recall: 0.877<br>F-score: 0.878 | Sentiment-based-features, semantic-features, uni-gram-features, and pattern-features | A pragmatic approach for HSD |
| **[18]** | 2018 | Neural Network and The LinearSVC | Italian Twitter and Facebook Posts | Precision: 0.81<br>Recall: 0.793<br>F-score: 0.80 | n-gram-based features | Supervised approaches to HSD |
| **[38]** | 2018 | RNNs | Short Hate Speech Messages From Twitter | Precision: 0.930<br>Recall: 0.930<br>F-score: 0.932 | BoW | Use of recurrent neural networks for HSD solution |
| **[39]** | 2018 | ANN (Backpropagation) | Indonesia Twitter Hate Speech Dataset | Accuracy: 0.8947<br>Precision: 0.8066<br>Recall: 0.9007 | TF-IDF | HSD using backpropagation neural networks |
| **[11]** | 2019 | Support Vector Machine, LSTM, Random Forest, Hierarchical attention-LSTM | Hate Speech Dataset in English-Hindi Languages | Accuracy: 0.70.7<br>Recall: 0.451<br>F-score: 0.487 | Word n-grams based features | HSD for English-Hindi languages |
| **[15]** | 2019 | Support Vector Machine, LSTM+MLP, Random Forest, Quadratic Discriminant Analysis and Linear Discriminant Analysis | HaterNet Hate Speech Twitter Dataset | Precision: 0.809<br>Recall: 0.789<br>F-score: 0.611<br>AUC: 0.828 | Frequency-based and embeddings-based features | HSD on Twitter using hybrid LSTM+MLP model |
| **[23]** | 2019 | Support Vector Machines | HatEval task dataset of SemEval-2019 | Macro-averaged F-score: 78.59 | Linguistic features and lexicons features | HSD on Twitter using machine learning approaches |
| **[26]** | 2019 | Logistic Regression and Bi-LSTM | Reddit and Facebook Hate Speech Dataset in Danish and English Languages | Macro-averaged F-score: 0.74 | Linguistic Features, GloVe, and word embeddings | HSD FOR Danısh language |
| **[40]** | 2019 | BERT+LSTM and BERT+CNN | Two Different Twitter Hate Speech Dataset | Precision: 0.81<br>Recall: 0.793<br>F-score: 0.80 | WordPiece tokenization | BERT + deep learning approaches for HSD |
| **[41]** | 2019 | ALBERTo+Bert classifier model | Italian Language from Twitter and Facebook Hate Speech Dataset | Macro-averaged F-score: 0.841 | SentencePiece tokenizer | HSD through ALBERTo |
| **[42]** | 2019 | Bi-LSTM models | Two Different English Hate Speech Dataset | Macro-averaged F-score: 0.78 | ELMo (word vector representation) | Transfer learning approach for HSD |
| **[43]** | 2019 | Support Vector Machine, Naive | Indonesia Twitter Hate | Averaged-Accuracy: | Word n-gram, ortography, | Multi-layer HSD with machine learning |

| Ref. No | Year | Classifiers/Algorithms /Frameworks/Tools/Models | Dataset | Obtained Metrics (max. values) | Used Feature Methods | Definition |
|---|---|---|---|---|---|---|
| **[31]** | 2017 | Support Vector Machine, Naive Bayes, k-Nearest Neighbors | Kenya Hate Speech Dataset | Accuracy: 0.7027<br>Precision: 0.70<br>Recall: 0.70<br>F-score: 0.70 | Uni-gram, bi-gram, and tri-gram features and TF-IDF | TF-IDF + n-grams based approach for HSD |
| **[32]** | 2017 | Twitter API Tool | Italian Twitter Corpus | Annotation category accuracy ratio: 0.54 | Cohen's kappa | Creating an HS dataset with Italian twitter corpus |
| **[33]** | 2017 | Support Vector Machine | German Hate Speech Twitter Corpus | Macro-averaged F-score: 0.65 | Uni-gram, bi-gram, and tri-gram features | Examining the effect of implicit on automatic HSD |
| **[34]** | 2017 | FastText Framework, CNNs and LSTMs. | Annotated tweets | Precision: 0.93<br>Recall: 0.93<br>F-score: 0.93 | TF-IDF and BoW | Deep learning approach for HSD |
| **[35]** | 2017 | Logistic Regression, Naive Bayes, Decision Trees, Random Forests, and LinearSVMs. | Hate speech and offensive language dataset | Precision: 0.91<br>Recall: 0.90<br>F-score: 0.90 | Lexical–based methods ( Uni-gram, bi-gram, and tri-gram features, TF-IDF) | Machine learning approach for HSD (Lexical-based) |
| **[36]** | 2018 | Logistic Regression, Multilayer Perceptron, LSTM, and CNN+GRU | Wikipedia and Twitter Hate Speech Dataset | F-score: 0.86 | n-gram features on both the character-level and word-level | HSD with character-level features and word-level features |
| **[37]** | 2018 | Random Forest, Support Vector Machine, and J48 | Three Different Twitter Hate Speech Dataset | Precision: 0.883<br>Recall: 0.877<br>F-score: 0.878 | Sentiment-based-features, semantic-features, uni-gram-features, and pattern-features | A pragmatic approach for HSD |
| **[18]** | 2018 | Neural Network and The LinearSVC | Italian Twitter and Facebook Posts | Precision: 0.81<br>Recall: 0.793<br>F-score: 0.80 | n-gram-based features | Supervised approaches to HSD |
| **[38]** | 2018 | RNNs | Short Hate Speech Messages From Twitter | Precision: 0.930<br>Recall: 0.930<br>F-score: 0.932 | BoW | Use of recurrent neural networks for HSD solution |
| **[39]** | 2018 | ANN (Backpropagation) | Indonesia Twitter Hate Speech Dataset | Accuracy: 0.8947<br>Precision: 0.8066<br>Recall: 0.9007 | TF-IDF | HSD using backpropagation neural networks |
| **[11]** | 2019 | Support Vector Machine, LSTM, Random Forest, Hierarchical attention-LSTM | Hate Speech Dataset in English-Hindi Languages | Accuracy: 0.70.7<br>Recall: 0.451<br>F-score: 0.487 | Word n-grams based features | HSD for English-Hindi languages |
| **[15]** | 2019 | Support Vector Machine, LSTM+MLP, Random Forest, Quadratic Discriminant Analysis and Linear Discriminant Analysis | HaterNet Hate Speech Twitter Dataset | Precision: 0.809<br>Recall: 0.789<br>F-score: 0.611<br>AUC: 0.828 | Frequency-based and embeddings-based features | HSD on Twitter using hybrid LSTM+MLP model |
| **[23]** | 2019 | Support Vector Machines | HatEval task dataset of SemEval-2019 | Macro-averaged F-score: 78.59 | Linguistic features and lexicons features | HSD on Twitter using machine learning approaches |
| **[26]** | 2019 | Logistic Regression and Bi-LSTM | Reddit and Facebook Hate Speech Dataset in Danish and English Languages | Macro-averaged F-score: 0.74 | Linguistic Features, GloVe, and word embeddings | HSD FOR Danısh language |
| **[40]** | 2019 | BERT+LSTM and BERT+CNN | Two Different Twitter Hate Speech Dataset | Precision: 0.81<br>Recall: 0.793<br>F-score: 0.80 | WordPiece tokenization | BERT + deep learning approaches for HSD |
| **[41]** | 2019 | ALBERTo+Bert classifier model | Italian Language from Twitter and Facebook Hate Speech Dataset | Macro-averaged F-score: 0.841 | SentencePiece tokenizer | HSD through ALBERTo |
| **[42]** | 2019 | Bi-LSTM models | Two Different English Hate Speech Dataset | Macro-averaged F-score: 0.78 | ELMo (word vector representation) | Transfer learning approach for HSD |
| **[43]** | 2019 | Support Vector Machine, Naive | Indonesia Twitter Hate | Averaged-Accuracy: | Word n-gram, ortography, | Multi-layer HSD with machine learning |

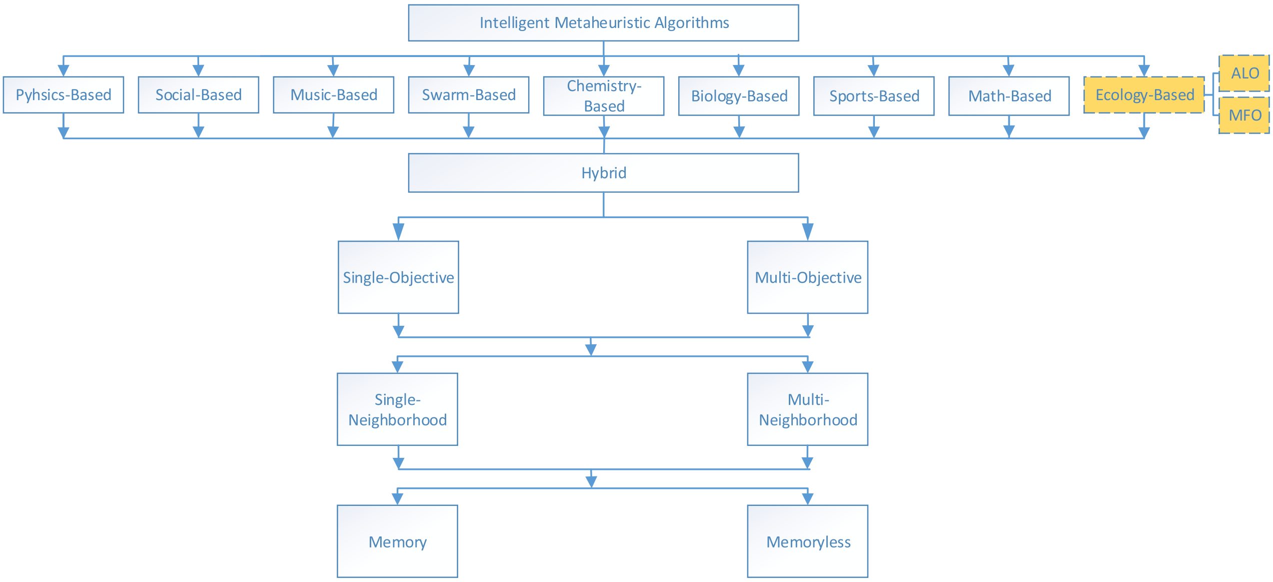


**FIGURE 1.** Intelligent metaheuristic optimization algorithms.

biology-based, math-based, ecology-based, etc. [53]. In fact, all of these algorithms have a common structure in performing [54]. These algorithms can be summarized as in Fig. 1. ALO and MFO algorithms based on ecology intelligence have recently been reported to work well for different problems [55], [56]. They have been evaluated to give better results for HSD in this paper. The biological structures and mathematical models of ALO and MFO algorithms are explained in the next subsection. SSO algorithm is a powerful algorithm used in recent studies, which is inspired by the intelligent behavior of spiders [57]–[59]. The other state-of-the-art TSA is bio-based algorithm inspired by the feeding behavior of turnicates [60]–[63].

### A. ANT LION OPTIMIZATION ALGORITHM (ALO)

Antlions are one of the insect groups in the Myrmeleontidae family. Their life cycle has two main phases: the larval stage and the adult stage. These creatures are often called “doodlebugs” because of the traces left in the sand when searching for a good place to set up traps. ALO was proposed inspiring by antlions hunting behavior. During the hunting process, the antlion absorbs the funnel pits into soft sand and then patiently waits at the bottom of the pit. Prey slides down and is quickly caught by the antlion. If prey tends to escape from the trap, the antlion throws sand at the edge of the trap to slide its victim towards the bottom of the pit. This causes the trap to collapse and the prey comes to the antlion [64].

There are many parameters in the mathematical modeling of ALO, which is developed inspired by these behaviors of ant lions and ants. Among these parameters, the random walks of ants while searching for foods in nature can be defined as in (1):

$$\begin{aligned} Xnn\,(t) = [0, cumsum\,(2r\,(t_1) - 1)\,, \\ \times\, cumsum\,(2r\,(t_2) - 1)\,, \ldots, cumsum(2r(t_n) - 1) \end{aligned} \quad (1)$$

Here *cumsum* calculates the cumulative total, *n* represents the maximum number of repetitions, *t* represents the steps of random walking, and $r(t)$ is the scholastic function is defined in (2):

$$r\,(t) = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & if\ rand \leq 0.5 \end{cases} \quad (2)$$

Here *t* is the step of random walking and *rand* is a random number generated with a uniform distribution in the interval [0, 1].

The position of the ants in the ALO algorithm is expressed in (3):

$$M_{ant} = \begin{matrix} A_{1,1} & A_{1,2} & \ldots & A_{1,d} \\ A_{2,1} & A_{2,2} & \ldots & A_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \ldots & A_{n,d} \end{matrix} \quad (3)$$

Here $M_{ant}$ is the position matrix of each ant, $A_{i,j}$ presents the value of *ith* variable for *jth* ant. *n* is the number of the ant, and *d* presents the number of variables. The fitness function of each ant is stored in the $M_{OA}$ matrix as in (4). In (4), *f* is the objective function.

$$M_{OA} = \begin{matrix} f\left(\left[A_{1,1}, A_{1,2}, \ldots, A_{1,d}\right]\right) \\ f\left(\left[A_{2,1}, A_{2,2}, \ldots, A_{2,d}\right]\right) \\ \vdots \\ f\left(\left[A_{n,1}, A_{n,2}, \ldots, A_{n,d}\right]\right) \end{matrix} \quad (4)$$

Equation (5), $M_{Antlion}$ is the matrix of each ant lion. $AL_{i,j}$ presents the current value of *ith* variable for *jth* antlion. *n* is the number of the antlion, and *d* presents the number of variables.

$$M_{Antlion} = \begin{matrix} AL_{1,1} & AL_{1,2} & \ldots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \ldots & AL_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ AL_{n,1} & AL_{n,2} & \ldots & AL_{n,d} \end{matrix} \quad (5)$$

Similarly, the $M_{OAL}$ matrix in (6) stores the fitness function of each ant lion:

$$M_{OAL} = \begin{matrix} f\left(\left[A_{1,1}, A_{1,2}, \ldots, A_{1,d}\right]\right) \\ f\left(\left[A_{2,1}, A_{2,2}, \ldots, A_{1,d}\right]\right) \\ \vdots \\ f\left(\left[A_{n,1}, A_{n,2}, \ldots, A_{n,d}\right]\right) \end{matrix} \tag{6}$$

Equation (7) is used to keep the ants walking randomly within the search area:

$$X_i^t = \frac{(X_i^t - a_i)(d_i - c_i^t)}{(d_i^t - a_i)} + c_i \tag{7}$$

$a_i$ presents minimum random walk of *ith* the variable. $c_i^t$ presents *ith* minimum value in *tth* iteration. $d_i^t$ presents *ith* maximum value in *tth* iteration.

Mathematical modeling of the capture of ants in ant lion's trap pits is given in (8) and (9):

$$c_i^t = Antlion_j^t + c^t \tag{8}$$

$$d_i^t = Antlion_j^t + d^t \tag{9}$$

Here $c_i^t$ is the minimum value of all variables in the *tth* iteration. $d_i^t$ is the maximum value of all variables in the *tth* iteration. Finally, $Antlion_j^t$ refers to the *jth* chosen position of the ant lion in the *tth* iteration.

The ant lion's hunting abilities are modeled by the roulette wheel selection. The mathematical model that explains how the trapped ant will slide towards the ant lion is given in (10) and (11) as follows:

$$c^t = \frac{c^t}{I} \tag{10}$$

$$d^t = \frac{d^t}{I} \tag{11}$$

$I$ ratio is calculated as in (12):

$$I = 10^w \frac{t}{T} \tag{12}$$

Here $t$ is the current iteration, $T$ is the maximum number of iterations, and $w$ is the constant depending on the current iteration and is defined in (13).

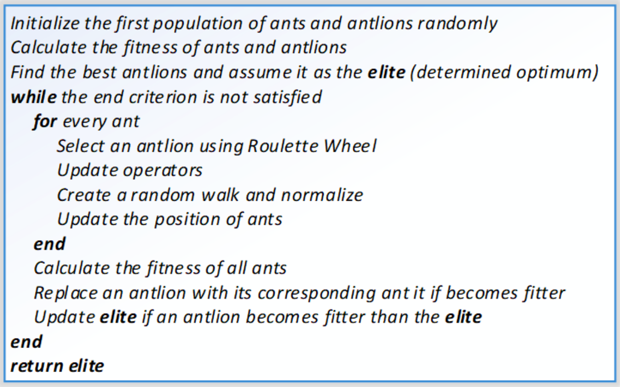
$$w = \begin{cases} 2 & if\ t > 0.1T \\ 3 & if\ t > 0.5T \\ 4 & if\ t > 0.75T \\ 5 & if\ t > 0.9T \\ 6 & if\ t > 0.95T \end{cases} \tag{13}$$

In each iteration, the best ant lion is considered the result. This means that each ant walks randomly around the selected antlion and is expressed in (14).

$$Ant_i^t = \frac{(R_A^t + R_E^t)}{2} \tag{14}$$

$R_A^t$ is expressed as the random walking of the ant-lion chosen by the roulette wheel in the *tth* iteration and $R_E^t$ is expressed as the best result walking randomly in the *tth* iteration.

The pseudo-code of ALO is illustrated in Fig. 2.



```
Initialize the first population of ants and antlions randomly
Calculate the fitness of ants and antlions
Find the best antlions and assume it as the elite (determined optimum)
while the end criterion is not satisfied
    for every ant
        Select an antlion using Roulette Wheel
        Update operators
        Create a random walk and normalize
        Update the position of ants
    end
    Calculate the fitness of all ants
    Replace an antlion with its corresponding ant it if becomes fitter
    Update elite if an antlion becomes fitter than the elite
end
return elite
```

**FIGURE 2.** The pseudo-code of ALO.

### B. MOTH-FLAME OPTIMIZATION ALGORITHM (MFO)

The MFO algorithm is a new method of metaheuristic optimization based on the simulation of specific navigation behavior of moths at night. Moths use a mechanism referred to as "transverse routing" for navigation. In this algorithm, a moth flies at a constant angle to the moon. It is a very effective method for long-distance travel on a straight road. Because the moon is far from the moth. That is, it guarantees that the moths fly along the straight line throughout the night. However, everyone usually observes that moths spiral around the lights. Because moths are deceived by artificial lights. Since these lights are extremely close to light sources, maintaining a constant angle to the light source causes moths to follow a spiral flight path [65].

In the mathematical model of the MFO algorithm, the set of moths is represented in an $M$ matrix. The matrix $M$ is shown in (15).

$$M_{months} = \begin{matrix} M_{1,1} & M_{1,2} & \ldots & M_{1,d} \\ M_{2,1} & M_{2,2} & \ldots & M_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1} & M_{n,2} & \ldots & M_{n,d} \end{matrix} \tag{15}$$

Here $n$ is the number of the moths, and $d$ presents the number of dimensions. For all moths, there is an $OM$ matrix to store fitness values. This matrix is illustrated in (16).

$$OM = \begin{matrix} f([OM_1]) \\ f([OM_2]) \\ \vdots \\ f([OM_n]) \end{matrix} \tag{16}$$

$n$ represents the number of months. The second important component in the algorithm is flames ($F$). The $F$ matrix can be considered a matrix similar to the moth matrix. The matrix $F$ is defined by (17).

$$F = \begin{matrix} F_{1,1} & F_{1,2} & \ldots & F_{1,d} \\ F_{2,1} & F_{2,2} & \ldots & F_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n,1} & F_{n,2} & \ldots & F_{n,d} \end{matrix} \tag{17}$$

It is also assumed that there is an *OF* matrix to store fitness values for flames. This matrix is defined in (18).

$$OF = \begin{matrix} f([OF_1]) \\ f([OF_2]) \\ \vdots \\ f([OF_n]) \end{matrix} \tag{18}$$

MFO algorithm is a triple function defined as in (19) to solve optimization problems:

$$M = (I, P.T) \tag{19}$$

Here *I* is the function that generates a random population of moths and the corresponding fitness values. The mathematical model of this function is as in (20):

$$I : \phi \rightarrow \{M, OM\} \tag{20}$$

The *P* function is the basic function for MFO. This function allows moths to be moved within the search area. This function takes the matrix *M* and returns the updated new *M* as in (21):

$$P : M \rightarrow M \tag{21}$$

The *T* function returns true if the termination criteria are met. If not provided, returns false. The *T* function is defined in (22).

$$T : M \rightarrow \{true, false\} \tag{22}$$

The general structure of the MFO algorithm created by *I*, *P*, and *T* functions can be summarized as in Table 2.

**TABLE 2. Creation of MFO algorithm with *I*, *P*, and *T* functions.**

```
$M = I()$;
    while $T(M)$ if equal to false
        $M = P(M)$;
    end
```

For the sake of mathematically simulating the behavior of moths, the position of each moth is updated using (23), taking into account a flame:

$$M_i = S\left(M_i, F_j\right) \tag{23}$$

In (23), $M_i$, $F_j$, and *S* respectively indicates the *ith* moth, *jth* flame, and spiral function. The following restrictions should be taken into account when creating the *S* function:

(1) The starting point of the spiral must start from the moths.
(2) The endpoint of the spiral should be the position of the flame.
(3) Fluctuation in the spiral range should not exceed the search area.

Considering these conditions, a logarithmic spiral is defined for the MFO algorithm as in (24):

$$S\left(M_i, F_j\right) = D_i.e^{b.t}\cos(2\pi t) + F_j \tag{24}$$

Here *D* denotes the distance of the *jth* moths for the *ith* flame. *b* is constant to describe the shape of the logarithmic spiral, and *t* is a random number between [−1, 1]. *D* is calculated as in (25).

$$D_i = \left|F_j - M_i\right| \tag{25}$$

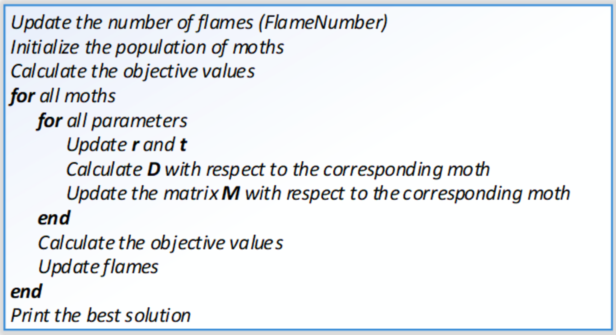
Equation (24) describes the spiral flight path of moths. From this equation, the next position of a moth is defined with respect to a flame. The parameter *t* in the spiral equation defines how close the next position of the moth must be to the flame.

Equation (26) is used to prevent the MFO algorithm from getting stuck to the local optimum.

$$flame\_no = round\left((N - I).\left(\frac{\mathrm{N} - 1}{T}\right)\right) \tag{26}$$

In (26), *I* indicate the current number of iterations. *N* and *T* represent the maximum number of moths and the maximum number of iterations, respectively.

The pseudo-code of the MFO algorithm, which is developed from the moths' behavior inspiration, is illustrated in Fig. 3.



```
Update the number of flames (FlameNumber)
Initialize the population of moths
Calculate the objective values
for all moths
    for all parameters
        Update r and t
        Calculate D with respect to the corresponding moth
        Update the matrix M with respect to the corresponding moth
    end
    Calculate the objective values
    Update flames
end
Print the best solution
```

**FIGURE 3. The pseudo-code of MFO.**

## IV. USED METHODOLOGIES

In this section, information about datasets, NLP steps, metaheuristic algorithms, and other algorithms was explained.

### A. COLLECTION OF DATA

In this study, three different experimental datasets were used. These experimental datasets constituted from real-world problems related to HSD. The data collected from Twitter and online web forums were pre-labeled. General characteristics of the data are listed in Table 3. These datasets that address the most important problems related to the real world. These issues are very important topics that directly target a person, formation, state structure, racism, and sexism. Detailed information about the datasets and the HSD study performed on the datasets were explained in section 5. These datasets with different characteristics were chosen because they were prepared to most suitable for the theme of the HSD problem.

**TABLE 3. Properties of the used datasets.**

| Ref. No | Names of DataSet | Contents | Languages | Size | Source | Text Type | Number of Labels | Direct Access Link |
|---|---|---|---|---|---|---|---|---|
| [66] | Dataset1 | Hate speech and offensive language | English | 25K | Twitter | Tweets | 3 | https://github.com/t-davidson/hate-speech-and-offensive-language |
| [35] | Dataset2 | Community-directed personal insults | English | 4K | Web Forum | Comments | 2 | https://www.kaggle.com/c/detecting-insults-in-social-commentary |
| [67] | Dataset3 | Hate speech against immigrants and women | English<br>Spanish | 13K | Twitter | Tweets | 2 | https://competitions.codalab.org/competitions/19935 |

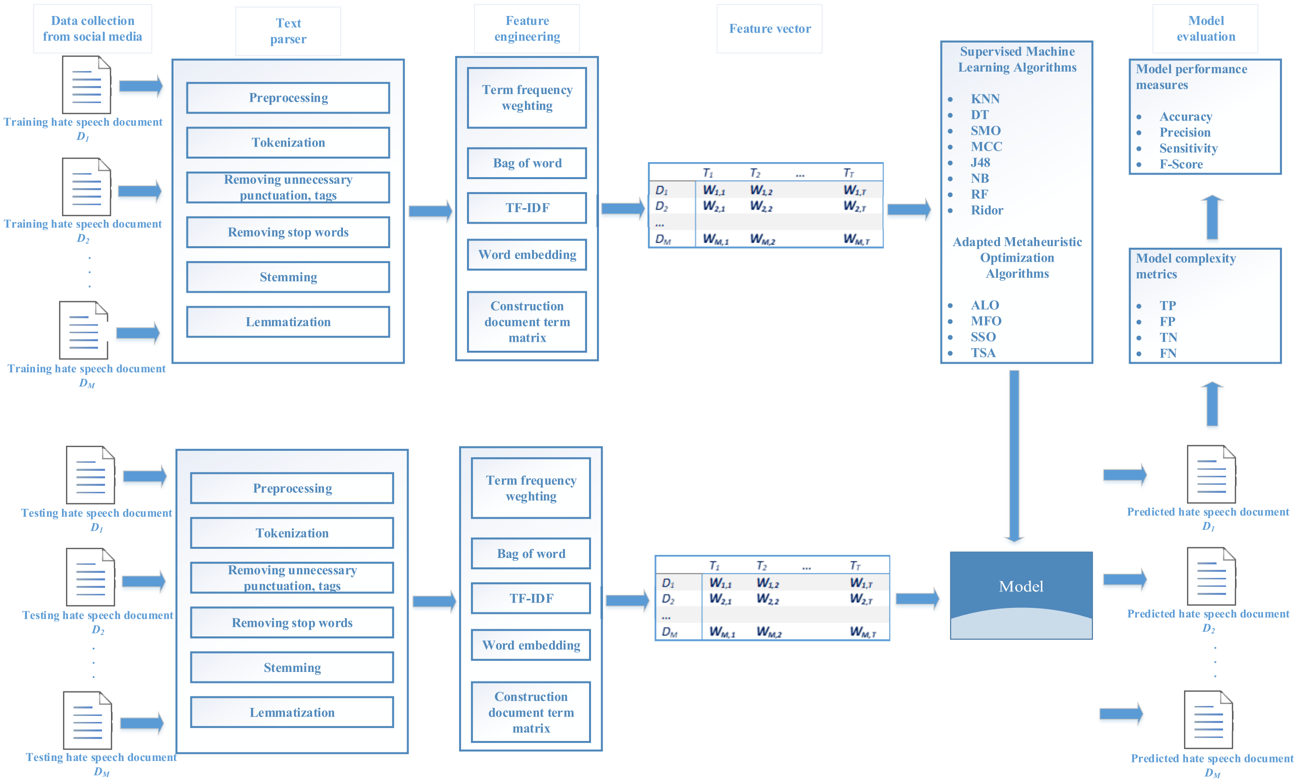

Data collection from social media
Text parser
Feature engineering
Feature vector
Supervised Machine Learning Algorithms
Model evaluation
Training hate speech document $D_1$
Training hate speech document $D_2$
Training hate speech document $D_M$
Preprocessing
Tokenization
Removing unnecessary punctuation, tags
Removing stop words
Stemming
Lemmatization
Term frequency weghting
Bag of word
TF-IDF
Word embedding
Construction document term matrix
$T_1$ $T_2$ ... $T_T$
$D_1$ $W_{1,1}$ $W_{1,2}$ $W_{1,T}$
$D_2$ $W_{2,1}$ $W_{2,2}$ $W_{2,T}$
...
$D_M$ $W_{M,1}$ $W_{M,2}$ $W_{M,T}$
KNN
DT
SMO
MCC
J48
NB
RF
Ridor
Adapted Metaheuristic Optimization Algorithms
ALO
MFO
SSO
TSA
Model performance measures
Accuracy
Precision
Sensitivity
F-Score
Model complexity metrics
TP
FP
TN
FN
Testing hate speech document $D_1$
Testing hate speech document $D_2$
Testing hate speech document $D_M$
Model
Predicted hate speech document $D_1$
Predicted hate speech document $D_2$
Predicted hate speech document $D_M$

**FIGURE 4. The structure of proposed HSD.**

### B. STEPS OF NLP AND DATA PRE-PROCESSING

In the pre-processing phase, our goal was to convert textual data to numerical form in order to create a document vector (Word2Vec). These steps are shown in Fig. 4. Before the document vector was created, many pre-processing steps had performed such as punctuation erasure, number filter, stopword filter, case converter, n-char filter, snowball stemmer, row filter, etc. BoW is an NLP method that extracts all word roots in each data. Word roots in each sentence were listed and the next step was taken. After the pre-processing steps was completed, tokenization was performed on a sentence basis. Then the most repetitive terms were calculated according to the Term Frequency (TF) value. Finally, a document vector was created. Thus, the feature extraction process representing datasets was completed. Then HSD problem became a classification problem. After the data are ready, other process steps are started with supervised machine learning and metaheuristic optimization algorithms. The proposed general structure of the HSD problem is demonstrated in Fig. 4.

As shown in Fig. 4, we split our datasets with 70% training and 30% test data. First, NLP steps were applied to de-noise on both training and test data. Then the term frequency value was calculated to extract the features that would represent the data. After the BoW process was completed, our data became a document matrix. Twelve algorithms were used in the classification stage. Four of them (ALO, MFO, SSO, and TSA) were metaheuristic algorithms. They were run in the fitness function designed for the HSD problem. The success of the algorithms was evaluated with the complexity calculation obtained on the test data. The adjustment of the fitness function modeled for the proper process of ALO, MFO, SSO, and TSA algorithms in HSD problem was explained in subtitles *D* and *E* in detail.

### C. SUPERVISED MACHINE LEARNING ALGORITHMS BASED ON ARTIFICIAL INTELLIGENCE

In order to compare the performance of metaheuristic optimization algorithms, various supervised machine learning algorithms were used for the HSD problem. Eight different supervised machine learning algorithms were used in the study. These algorithms have higher success in classification problems. Moreover, they are the most widely used algorithms in the literature. k-Nearest Neighbor algorithm (KNN), Decision Table (DT), Sequential Minimal Optimization (SMO), Multi-Class Classification (MCC), J48, Naive Bayes (NB), Random Forest (RF), and Ridor algorithms were used in this study.

### D. MODELING OF HSD WITH OPTIMIZATION ALGORITHMS

In the modeling stage of the optimization algorithms for the HSD problem, it was necessary to set the initial parameters of the algorithms firstly. The values of the initial parameters of the ALO, MFO, SSO, and TSA algorithms are listed in Table 4.

**TABLE 4.** **Initial parameters of the ALO, MFO, SSO, and TSA algorithms.**

| |
|---|
| Dimension (*d*) = number of features obtained for each dataset |
| Lower bound (*lb*) = 0 |
| Upper bound (*ub*) = 1 |
| Maximum number of iterations = 1000 |
| Number of populations = 30 |
| Number of experiments = 10 |

The total number of experiments for each dataset was determined as 10. The maximum number of iterations was 1000 for each experiment. At the end of each experiment, the measurement criteria with the best fitness value were recorded. The final measurement criteria results were obtained by taking the average of the best fitness function results of 10 experiments. The population size for the ALO, MFO, SSO, and TSA algorithms was 30. Each individual was encoded by real numbers between 0 and 1. These reel numbers that form the search agents were updated at the end of the next iteration. The individual with the best fitness was recognized as the best solution for the existing iteration.

### E. MODELING OF FITNESS FUCTION

While preparing the fitness function, each of the measurement criteria was designed to have a weight. Measurement criteria such as accuracy, precision, sensitivity, and f-score were used to define the fitness function for the HSD problem. The fitness function proposed for the HSD problem is defined in (27)

$$fitnessn\,function = a_1.accuracy + a_2.precision + a_3.sensitivity + a_4.f - score \quad (27)$$

where $a_1$, $a_2$, $a_3$, and $a_4$ variables were random weights. The sum of these variables must be equal to 1 in the fitness function. One can easily give importance by using the relevant weights for the metrics. Furthermore, metrics or objectives can be easily removed from or integrated into this fitness function. That is why, it seems flexible.

After several test runs, the weight coefficients were assigned as 0.4, 0.2, 0.2, and 0.2 for the values $a_1$, $a_2$, $a_3$, and $a_4$, respectively. While designing the fitness function, our aim was to create the most suitable candidate solution to represent each class in our datasets. The success of the best fit candidate solution obtained was run on test data. This fitness function is flexible for the HSD problem. It was analyzed as a result of the experiments that it obtained good results. Different constraints and parameters can be easily integrated into this fitness function. For example; AUROC, FPR, etc. criteria can be integrated. In addition, the fitness function can be updated by giving a weight in the fitness function in the TP and FP values of the complexity metrics to be calculated. In this sense, it is flexible.

The ALO, MFO, SSO, and TSA algorithms aim to find the most suitable models to represent the HSD problem in all training datasets. After determining the most suitable candidate solutions representing the hate and not hate data in the HSD problem, the success of the developed model on the test data was implemented. Using the Jaccard similarity on the test data, the test data were labeled in which way it was closer to the candidate solution. Jaccard similarity is defined as in (28).

$$jaccard\_similarity_{ij} = \left| \frac{X_i \cap D_j}{X_i \cup D_j} \right| \quad (28)$$

Here, $X_i = X_{i1}, X_{i2}, \ldots, X_{iM}$ represents the *ith* individual (antlion, moth, spider, or turnicate) in ALO, MFO, SSO, and TSA algorithms. $D_j = D_{j1}, D_{j2}, \ldots, D_{jM}$ represents the *jth* data in the document vector created after the NLP steps. Equation (29) can be used to summarize the relationship between the *X* and *D* vectors.

$$jaccardn_{similarity(X_i,D_j)} = \frac{\sum_{k=1}^{M} X_{ik} D_{jk}}{\sum_{k=1}^{M} X_{ik}^2 + \sum_{k=1}^{M} D_{jk}^2 - \sum_{k=1}^{M} X_{ik} D_{jk}} \quad (29)$$

In another important process step, complexity metrics were calculated for the prediction generated using Jaccard similarity. In order to measure model success, performance measurement criteria (accuracy, precision, sensitivity, and f-score values) were calculated by using complexity metrics. Performance measurement criteria and complexity metrics are summarized in Table 5.

**TABLE 5. Performance measurement criteria and complexity metrics.**

| Criteria | Explanation | Complexity metrics |
|---|---|---|
| Accuracy | $\frac{(TP + TN)}{(TP + TN + FP + FN)}$ | **TP:** True Positive<br>**TN:** True Negative<br>**FP:** False Positive<br>**FN:** False Negative |
| Precision | $\frac{(TP)}{(TP + FP)}$ | |
| Sensitivity | $\frac{(TP)}{(TP + FN)}$ | |
| F-score | $2 . \frac{(TP)}{(TP + FP + FN)}$ | |

In order to solve the HSD problem, properly modeled and adapted metaheuristic optimization methods were run on datasets. In the next section, the experimental results obtained were examined.

## V. EXPERIMENTS AND RESULTS

In the experiments, datasets containing three different real-world HSD problems were used. In all experiments, 70% of the dataset was used for training and the remaining 30% was used for testing. All the algorithms were run under equal conditions. Used computer features are as follows: Intel Core i5 10210U processor, 8GB RAM, 256GB SSD, and 4GB MX110 graphics card.

### A. RESULTS FOR HATE SPEECH AND OFFENSIVE LANGUAGE

Dataset1 consists of tweets containing about 25K hate speech and offensive language obtained from Twitter. There are three classes in this dataset. These are hate speech, offensive language, and neither [66]. The language of the dataset is English. BoW was applied in the English language dataset.

After the pre-processing steps, 48 features were obtained on this dataset. The dimension variable of ALO, MFO, SSO, and TSA algorithms was set to 48. 30 new candidates were generated in each iteration. Then one of them was accepted as the candidate model with the highest value in the fitness function. Other parameters remained constant. The performance comparison using ALO, MFO, SSO, TSA, and eight different supervised machine learning algorithms is demonstrated in Table 6.

The highest accuracy value (92.1%) was obtained by the ALO algorithm. The highest precision value was achieved by SSO algorithms. ALO was ranked sixth in the precision criteria. ALO algorithm again prevailed the highest sensitivity value (91.9%). TSA acquired the highest f-score values. ALO was able to find a place in the second to last place for the f-score criteria.

The second highest accuracy measurement criteria was yielded by TSA. With an accuracy value of 90.7%, the MFO algorithm ranked third for the first dataset. This algorithm had the third highest precision (88.4%). In the sensitivity criterion, the MFO and TSA acquired the second highest value (90.8%) just behind the ALO algorithm. In the f-score criterion, the MFO algorithm attained the same success as the SMO and DT algorithms with a value of 87.7%. Since the covariance and speed of convergence of the ALO algorithm were high, it successfully surpassed the MFO algorithm. Furthermore, ALO prevailed higher performance because it worked more effectively in terms of the search strategy. The J48 algorithm achieved the worst value for this dataset in almost all performance measurement criteria. DT only obtained the lowest precision. Fig. 5 shows the performance of the algorithms with a bar graph.

### B. RESULTS FOR COMMUNITY-DIRECTED PERSONAL INSULT

The dataset2 consists of approximately 4K comments containing personal insults. These comments were collected from online web forum pages. This dataset consists of two classes: insult and not insult [35]. The language of the data collected from public web forum pages is English.

The 27 features for this dataset were obtained after pre-processing and feature extraction. Therefore, the dimension value of ALO, MFO, SSO, and TSA algorithms was set to 27. Other evolutionary parameters of ALO, MFO, SSO, and TSA were not changed.

The performance values of the experimental study using twelve different algorithms are given in Table 7.

ALO algorithm attained the second highest accuracy value for dataset2. It prevailed the highest value with a value of 73.0% in the precision value, which is another performance measurement metric. For the sensitivity value, the DT algorithm outperformed the ALO algorithm, so the third highest sensitivity value was obtained by ALO. TSA prevailed the first place with 73.1% f-score value.

MFO, SSO, and TSA ranked themselves in 3rd place with 72.9% accuracy for dataset2. MFO and TSA; with a precision of 72.3%, ranked fifth behind ALO, DT, SSO, and KNN algorithms. The MFO algorithm reached the sixth and eighth ranked in sensitivity and f-score values, respectively. The algorithm acquired higher sensitivity and f-score values from Ridor, RF, and KNN algorithms. ALO passed MFO for this dataset in performance success.

The highest accuracy and sensitivity metric values were attained by DT algorithm. The best f-score value was acquired by TSA. In addition, DT and ALO algorithms were reached the same f-score value (71%). However, with a slight difference, the DT algorithm prevailed with high values in accuracy and sensitivity values. Also, the RF algorithm was observed to be bad for this dataset. All algorithms and comparison metrics are illustrated in Fig. 6. Accuracy and sensitivity values were appeared to be very close to each other.

### C. RESULTS FOR HATE SPEECH AGAINST IMMIGRANTS AND WOMEN

Dataset3 consists of about 13K tweets. This dataset, which includes hate speech against women and migration, consists

**TABLE 6.** Experiment results for dataset1.

| Dataset1 | | Feature Model | Performance Measurement Criteria | | | |
|---|---|---|---|---|---|---|
| | | | Accuracy | Precision | Sensitivity | F-Score |
| Used Algorithms | **ALO** | BoW+TF +Word2Vec | **0.921** | 0.879 | **0.919** | 0.875 |
| | **MFO** | BoW+TF+Word2Vec | 0.907 | 0.884 | 0.908 | 0.877 |
| | **SSO** | BoW+TF+Word2Vec | 0.898 | **0.886** | 0.898 | 0.878 |
| | **TSA** | BoW+TF+Word2Vec | 0.920 | 0.885 | 0.908 | **0.895** |
| | **KNN *(k=5)*** | BoW+TF+Word2Vec | 0.895 | 0.885 | 0.896 | 0.882 |
| | **KNN *(k=3)*** | BoW+TF+Word2Vec | 0.897 | 0.885 | 0.898 | 0.883 |
| | **DT** | BoW+TF+Word2Vec | 0.894 | 0.869 | 0.893 | 0.877 |
| | **SMO** | BoW+TF+Word2Vec | 0.898 | 0.876 | 0.899 | 0.877 |
| | **MCC** | BoW+TF+Word2Vec | 0.897 | 0.879 | 0.897 | 0.883 |
| | **J48** | BoW+TF+Word2Vec | 0.892 | 0.876 | 0.891 | 0.873 |
| | **NB** | BoW+TF+Word2Vec | 0.873 | 0.880 | 0.873 | 0.876 |
| | **RF** | BoW+TF+Word2Vec | 0.893 | 0.878 | 0.894 | 0.882 |
| | **Ridor** | BoW+TF+Word2Vec | 0.893 | 0.881 | 0.894 | 0.876 |

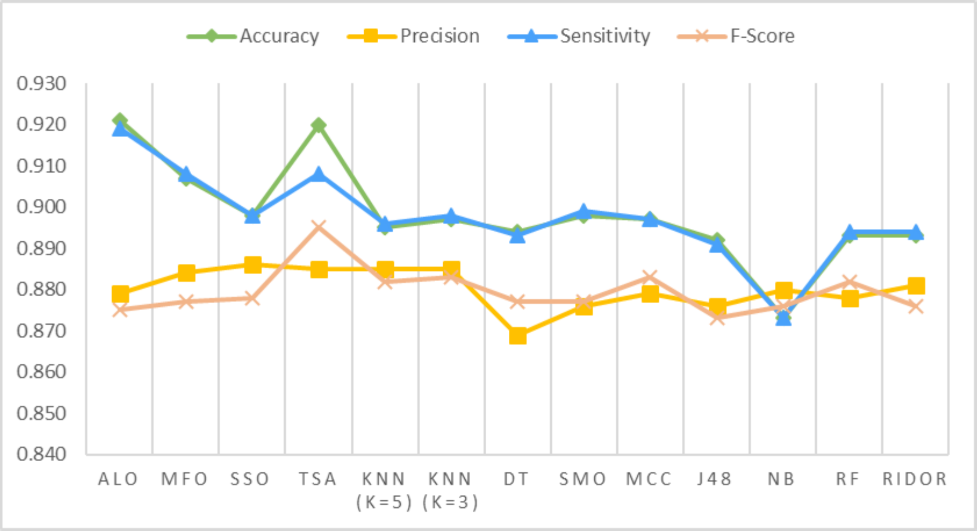


**FIGURE 5.** Performances of the algorithms for dataset1.

**TABLE 7.** Experiment results for dataset2.

| Dataset2 | | Feature Model | Performance Measurement Criteria | | | |
|---|---|---|---|---|---|---|
| | | | Accuracy | Precision | Sensitivity | F-Score |
| Used Algorithms | **ALO** | BoW+TF +Word2Vec | 0.730 | **0.730** | 0.730 | 0.710 |
| | **MFO** | BoW+TF+Word2Vec | 0.729 | 0.723 | 0.728 | 0.699 |
| | **SSO** | BoW+TF+Word2Vec | 0.729 | 0.724 | 0.731 | 0.703 |
| | **TSA** | BoW+TF+Word2Vec | 0.729 | 0.723 | 0.731 | **0.731** |
| | **KNN *(k=5)*** | BoW+TF+Word2Vec | 0.719 | 0.724 | 0.720 | 0.673 |
| | **KNN *(k=3)*** | BoW+TF+Word2Vec | 0.716 | 0.712 | 0.715 | 0.674 |
| | **DT** | BoW+TF+Word2Vec | **0.733** | 0.725 | **0.732** | 0.708 |
| | **SMO** | BoW+TF+Word2Vec | 0.724 | 0.712 | 0.725 | 0.704 |
| | **MCC** | BoW+TF+Word2Vec | 0.729 | 0.721 | 0.730 | 0.704 |
| | **J48** | BoW+TF+Word2Vec | 0.717 | 0.705 | 0.718 | 0.705 |
| | **NB** | BoW+TF+Word2Vec | 0.715 | 0.708 | 0.716 | 0.710 |
| | **RF** | BoW+TF+Word2Vec | 0.699 | 0.682 | 0.699 | 0.678 |
| | **Ridor** | BoW+TF+Word2Vec | 0.717 | 0.704 | 0.718 | 0.695 |

of two classes. The main source of these data is Twitter, which is one of the popular online social networks [67]. These tweets would be labeled as hate-full if there was hate towards women and immigrants. Otherwise, they were labeled as not-hate-full.

This dataset consist of English and Spanish tweets. However, BoW was performed in accordance with the English language and the Spanish tweets were excluded.

A total of 40 features was extracted for this dataset3. The dimension variable of ALO, MFO, SSO, and TSA algorithms was set to 40. Other evolutionary parameters of the meta-heuristic ALO, MFO, SSO, and TSA algorithms remained stable. The success of the algorithms used in the study is demonstrated in Table 8.

ALO prevailed the leading in performance competition for all measurement criteria. It was seen that ALO was the

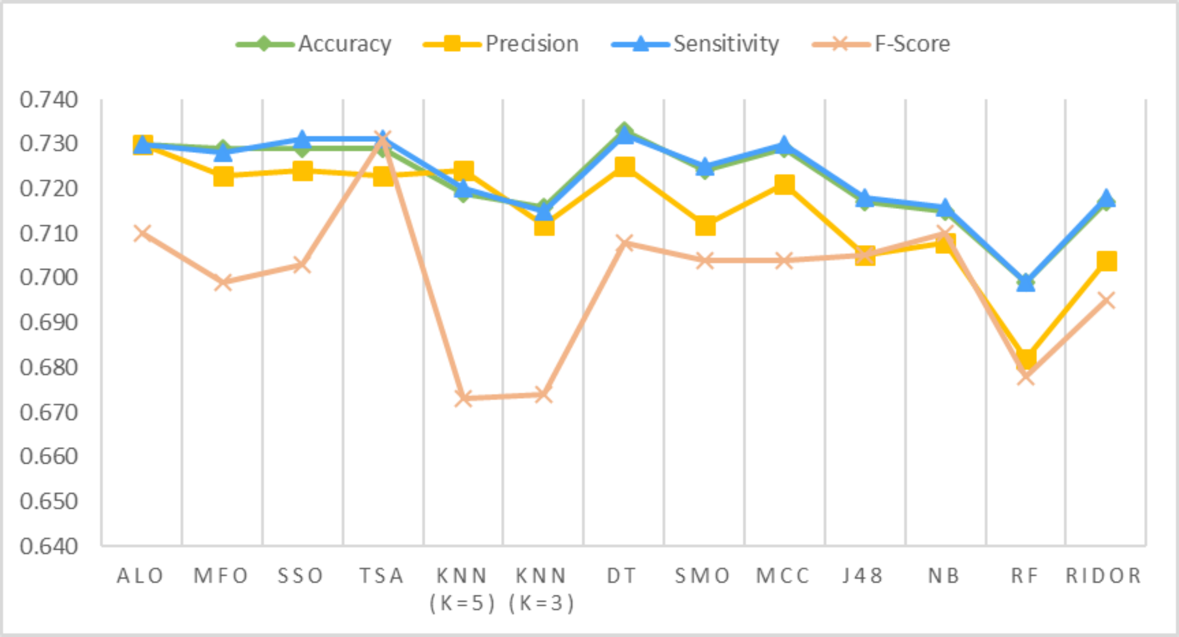


**FIGURE 6.** Performances of the algorithms for Dataset2.

**TABLE 8.** Experiment results for dataset3.

| | Dataset3 | Feature Model | Performance Measurement Criteria | | | |
|---|---|---|---|---|---|---|
| | | | Accuracy | Precision | Sensitivity | F-Score |
| **Used Algorithms** | **ALO** | BoW+TF +Word2Vec | **0.721** | **0.707** | **0.719** | **0.701** |
| | **MFO** | BoW+TF+Word2Vec | 0.691 | 0.688 | 0.679 | 0.677 |
| | **SSO** | BoW+TF+Word2Vec | 0.690 | 0.677 | 0.689 | 0.675 |
| | **TSA** | BoW+TF+Word2Vec | 0.717 | 0.700 | 0.718 | 0.700 |
| | **KNN *(k=5)*** | BoW+TF+Word2Vec | 0.669 | 0.666 | 0.670 | 0.666 |
| | **KNN *(k=3)*** | BoW+TF+Word2Vec | 0.671 | 0.669 | 0.672 | 0.669 |
| | **DT** | BoW+TF+Word2Vec | 0.643 | 0.663 | 0.644 | 0.646 |
| | **SMO** | BoW+TF+Word2Vec | 0.668 | 0.668 | 0.668 | 0.668 |
| | **MCC** | BoW+TF+Word2Vec | 0.669 | 0.667 | 0.670 | 0.668 |
| | **J48** | BoW+TF+Word2Vec | 0.678 | 0.677 | 0.679 | 0.677 |
| | **NB** | BoW+TF+Word2Vec | 0.660 | 0.674 | 0.660 | 0.662 |
| | **RF** | BoW+TF+Word2Vec | 0.673 | 0.672 | 0.674 | 0.672 |
| | **Ridor** | BoW+TF+Word2Vec | 0.666 | 0.702 | 0.667 | 0.666 |

outperforming algorithm for this dataset with its success. It attained an accuracy of 72.1% with the advantage of the search strategy. It surpassed its competitors in other benchmarks.

The second-best accuracy, and f-score values were attained by the TSA. The second best precision value (70.2%) was acquired by the Ridor algorithm. MFO managed to outpace its competitors. DT was the worst-performing algorithm for this dataset. It was ranked last in all performance criteria. With an accuracy of 64.3%, it was far behind its competitors. ALO and MFO succeeded in leaving many algorithms behind with their success in solving HSD problems. The performance graph of all algorithms is demonstrated in Fig. 7.

As can be seen from the results of all experiments, used metaheuristic optimization algorithms to solve the HSD problem were provided more success than those of classical supervised machine learning algorithms. The results of the experiments showed that metaheuristic algorithms can be used in the solution of many different social network and media problems.

As the number of data increased, the running time of the metaheuristic algorithms applied for the experiments also took time. One of the limit to the operation of algorithms is the excessive number of data. In addition, the unbalanced number of data belonging to each class in the datasets negatively affected the high performance of the algorithms. Better performance would have been achieved if there were datasets with balanced distribution. Furthermore, determining the algorithm parameters a priori and stochastic behaviours seem another limitations. To prove the reliability of the algorithms, each algorithm was run 10 times for all datasets. Measurement criteria were calculated by taking the average of 10 runs.

**TABLE 9.** Comparison of studies.

| Studies | Dataset | Feature Model | Classifier | Accuracy | Precision | Sensitivity | F-Score |
|---|---|---|---|---|---|---|---|
| **[35]** | Dataset1 | Lexical–based methods (Uni-gram, bi-gram, and tri-gram features, TF-IDF) | Logistic Regression, Naive Bayes, Decision Trees, Random Forests, and LinearSVMs. | - | 0.91 (max.) | 0.90 (max.) | 0.90 (max.) |
| **[40]** | Dataset1 | WordPiece tokenization | BERT | - | 0.91 | 0.91 | 0.91 |
| | | | BERT+Nonlinear Layers | - | 0.76 | 0.78 | 0.77 |
| | | | BERT+LSTM | - | 0.91 | **0.92** | **0.92** |
| | | | BERT+CNN | - | **0.92** | **0.92** | **0.92** |
| **[47]** | Dataset1 | GloVe and TF-IDF | Support Vector Machines, Naive Bayes, Logistic Regression, Decision Tree, Random Forest, CNN, LSTM, BiLSTM | 0.752 (max.) | - | - | - |
| **[50]** | Dataset1 | Pre-trained toxicity representation | LSTM | - | - | - | 0.892 |
| | | | LSTM+HATE-GAN | - | - | - | 0.896 |
| | | | CNN | - | - | - | 0.89 |
| | | | CNN+HATE-GAN | - | - | - | 0.895 |
| | | | CNN+LSTM | - | - | - | 0.887 |
| | | | CNN+LSTM+HATE-GAN | - | - | - | 0.894 |
| **[51]** | Dataset3 | Node2Vec | FastText+LSTM | - | - | - | 0.591 |
| | | | Node2Vec+LSTM | - | - | - | 0.548 |
| | | | KGs | - | - | - | 0.618 |
| **This Study** | Dataset1 | BoW+TF +Word2Vec | ALO | **0.921** | 0.879 | 0.919 | 0.875 |
| | | | MFO | 0.907 | 0.884 | 0.908 | 0.877 |
| | Dataset3 | BoW+TF +Word2Vec | ALO | **0.721** | **0.707** | **0.719** | **0.701** |
| | | | MFO | 0.691 | 0.688 | 0.679 | 0.677 |

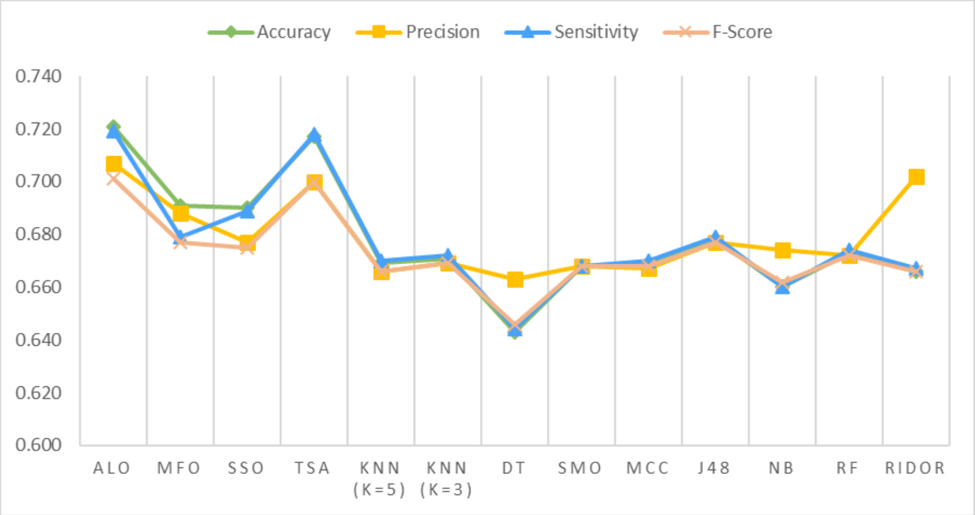


**FIGURE 7.** Performances of the algorithms for Dataset3.

It is predicted that the improved versions of the applied and proposed state-of-the-art algorithms will produce promising results in solving HSD and other social media and network problems.

The experimental results in this study and the results obtained from the previous studies using the same dataset are compared in Table 9. It was seen that the best result for Dataset1 was obtained by the ALO algorithm.

**TABLE 10.** Performance ranking for all datasets.

| | | |
|---|---|---|
| **Dataset1** | **Accuracy** | **ALO** > TSA > **MFO** > SSO = SMO > KNN *(k=3)* = MCC > KNN *(k=5)* > DT > RF = Ridor > J48 > NB |
| | **Precision** | SSO > KNN *(k=3)* = KNN *(k=5)* = TSA > **MFO** > Ridor > NB > **ALO** = MCC > RF > SMO = J48 > DT |
| | **Sensitivity** | **ALO** > **MFO** = TSA > SMO > SSO = KNN *(k=3)* > MCC > KNN *(k=5)* > RF = Ridor > DT > J48 > NB |
| | **F-Score** | TSA > KNN *(k=3)* = MCC > KNN *(k=5)* = RF > SSO > **MFO** = SMO = DT > Ridor = NB > **ALO** > J48 |
| **Dataset2** | **Accuracy** | DT > **ALO** > **MFO** = SSO = TSA = MCC > SMO > KNN *(k=5)* > Ridor = J48 > KNN *(k=3)* > NB > RF |
| | **Precision** | **ALO** > DT > SSO = KNN *(k=5)* > **MFO** = TSA > MCC > SMO = KNN *(k=3)* > NB > J48 > Ridor > RF |
| | **Sensitivity** | DT > SSO = TSA > **ALO** = MCC > **MFO** > SMO > KNN *(k=5)* > Ridor = J48 > NB > KNN *(k=3)* > RF |
| | **F-Score** | TSA > **ALO** = NB > DT > J48 > MCC = SMO > SSO > **MFO** > Ridor > RF > KNN *(k=3)* > KNN *(k=5)* |
| **Dataset3** | **Accuracy** | **ALO** > TSA > **MFO** > SSO > J48 > RF > KNN *(k=3)* > KNN *(k=5)* = MCC > SMO > Ridor > NB > DT |
| | **Precision** | **ALO** > Ridor > TSA > **MFO** > SSO = J48 > NB > RF > KNN *(k=3)* > SMO > MCC > KNN *(k=5)* > DT |
| | **Sensitivity** | **ALO** > TSA > SSO > **MFO** = J48 > RF > KNN *(k=3)* > KNN *(k=5)* = MCC > SMO > Ridor > NB > DT |
| | **F-Score** | **ALO** > TSA > **MFO** = J48 > SSO > RF > KNN *(k=3)* > SMO = MCC > KNN *(k=5)* = Ridor > NB >DT |

In addition, it was concluded that the highest values were obtained by ALO and MFO in all complexity criteria for Dataset3.

## VI. CONCLUSION AND RECOMMENDATION

This section is divided into two sections as conclusions and recommendations. In these subsections, the importance and success of the experimental studies are discussed. The novelty and precious of the study are particularly emphasized. It is anticipated that it will bring a new breath to the literature.

### A. CONCLUSION

The HSD problem, which is a major threat in social networks, is the subject of this study. Hate speech or defamatory messages targeting a community or group and quickly posted on social networks should be detected before they reach more users. This study proposes two new optimization-based approaches to solve the HSD problem in online social networks. ALO and MFO, which are the most recent metaheuristic algorithms, were adapted for the first time in the literature to solve the HSD problem. Eight different supervised machine learning algorithms, SSO, and state-of-the-art TSA were used to compare the performance of the proposed metaheuristic based approaches. For the selected real-world problems related to HSD, the pre-processing phase was completed with NLP methods. Feature extraction was performed by using BoW+TF+Word2Vec methods together. Then twelve different algorithms competed to solve HSD problems. In the experiments, except for one dataset, the highest accuracy, precision, sensitivity, and f-score values are obtained from the adapted ALO algorithm. MFO and TSA algorithms followed the ALO algorithm in the performance race with respect to related metrics. ALO achieved higher performance because it worked more effectively in terms of the search strategy. Since the covariance and speed of convergence of the ALO algorithm were high, it successfully surpassed the MFO algorithm. These two algorithms are successful and promising to solve HSD and seem alternative solution search methods for other social networking problems.

Metaheuristic optimization algorithms, which are capable of efficiently solving many real-world problems, achieved better performance for the HSD problem in terms of many metrics. This study is a reference study to solve HSD problems with the optimization approach. The success of this study shows that the use of this optimization-based approach in other social networking problems will increase. While other supervised machine learning methods show the black-box approach, the optimization-based approach can be adjusted and used according to many problems with its flexible fitness function. Furthermore, the fitness function can be easily adjusted to different datasets for different complex social media problems.

### B. RECOMMENDATIONS

In addition to the methods used in this study, more successful results can be obtained by using different approaches. These can be listed as follows;

(1) Different similarity measurement methods (cosine, dice, etc.) can be used instead of the Jaccard similarity.
(2) In the binary conversion process in representation schemes, different conversion methods can be proposed and used.
(3) Using the Pareto approach, the performance of ALO and MFO algorithms can be increased by handling many objectives for the HSD problem.
(4) Hybrid or adaptive methods may be proposed to improve the performance of optimization algorithms for this problem.
(5) These optimization methods can be integrated into the classical machine learning algorithms for better results in the automatic HSD problem.
(6) For the solution of the HSD problem, parameter optimization can be performed using metaheuristic optimization algorithms in deep learning algorithms.
(7) Parallel or distributed versions of these methods can be proposed for efficient results in big data with respect to many metrics.

## REFERENCES

[1] A. A. Aziz and A. Starkey, "Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches," *IEEE Access*, vol. 8, pp. 17722–17733, 2020.

[2] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, "Fake news stance detection using deep learning architecture (CNN-LSTM)," *IEEE Access*, vol. 8, pp. 156695–156706, 2020.

[3] H. Bingol and B. Alatas, "Rumor detection in social media using machine learning methods," in *Proc. 1st Int. Informat. Softw. Eng. Conf. (UBMYK)*, Nov. 2019, pp. 1–4.

[4] E. V. Altay and B. Alatas, "Detection of cyberbullying in social networks using machine learning methods," in *Proc. Int. Congr. Big Data, Deep Learn. Fighting Cyber Terrorism (IBIGDELFT)*, Dec. 2018, pp. 87–91.

[5] C. Baydogan and B. Alatas, "Detection of customer satisfaction on unbalanced and multi-class data using machine learning algorithms," in *Proc. 1st Int. Informat. Softw. Eng. Conf. (UBMYK)*, Nov. 2019, pp. 1–5.

[6] I. Makarov, O. Gerasimova, P. Sulimov, and L. E. Zhukov, "Dual network embedding for representing research interests in the link prediction problem on co-authorship networks," *PeerJ Comput. Sci.*, vol. 5, p. e172, Jan. 2019.

[7] V. Sharma, V. K. Jain, and A. Kumar, *An Introduction to Optimization Techniques*. Boca Raton, FL, USA: CRC Press, 2021.

[8] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13170–13180, 2011, doi: 10.1016/j.eswa.2011.04.126.

[9] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Comput. Ind. Eng.*, vol. 158, Aug. 2021, Art. no. 107408, doi: 10.1016/j.cie.2021.107408.

[10] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired Metaheuristic algorithm for global optimization problems," *Int. J. Intell. Syst.*, pp. 1–72, 2021, doi: 10.1002/int.22535.

[11] T. Y. S. S. Santosh and K. V. S. Aravind, "Hate speech detection in hindi-english code-mixed social media text," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2019, pp. 310–313.

[12] R. Kshirsagar, T. Cukuvac, K. McKeown, and S. McGregor, "Predictive embeddings for hate speech detection on Twitter," 2018, *arXiv:1809.10644*. [Online]. Available: http://arxiv.org/abs/1809.10644

[13] Z. Waseem, "Are you a racist or am i seeing things? Annotator influence on hate speech detection on Twitter," in *Proc. 1st Workshop NLP Comput. Social Sci.*, 2016, pp. 138–142.

[14] A. Arango, J. Pérez, and B. Poblete, "Hate speech detection is not as easy as you may think: A closer look at model validation," in *Proc. 42nd Int. ACM sigir Conf. Res. Develop. Inf. Retr.*, 2019, pp. 45–54.

[15] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, "Detecting and monitoring hate speech in Twitter," *Sensors*, vol. 19, no. 21, p. 4654, 2019, doi: 10.3390/s19214654.

[16] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," 2017, *arXiv:1712.06427*. [Online]. Available: http://arxiv.org/abs/1712.06427

[17] S. Biere, S. Bhulai, and M. B. Analytics, "Hate speech detection using natural language processing techniques," M.S. thesis, Dept. Math., Fac. Sci., Master Bus. Anal., Vrije Univ. Amsterdam, Amsterdam, The Netherlands, 2018.

[18] C. Michele, M. Stefano, A. Pinar, R. Sprugnoli, C. Elena, T. Sara, and V. Serena, "Comparing different supervised approaches to hate speech detection," in *Proc. EVALITA*, 2018, pp. 230–234.

[19] D. Robinson, Z. Zhang, and J. Tepper, "Hate speech detection on Twitter: Feature engineering vs feature selection," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2018, pp. 46–49.

[20] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proc.2nd Workshop Lang. Social Media*, 2012, pp. 19–26.

[21] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in *Proc. NAACL Student Res. Workshop*, 2016, pp. 88–93.

[22] V. Santucci, S. Spina, A. Milani, G. Biondi, and G. Di Bari, "Detecting hate speech for Italian language in social media," in *Proc. 5th Italian Conf. Comput. Linguistics, (CLiC-it) EVALITA*, vol. 2263, 2018, pp. 1–6.

[23] N. Bauwelinck and E. Lefever, "Measuring the impact of sentiment for hate speech detection on Twitter," in *Proc. HUSO 5th Int. Conf. Hum. Social Anal., (IARIA)*, 2019, pp. 17–22.

[24] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PLoS ONE*, vol. 14, no. 8, Aug. 2019, Art. no. e0221152.

[25] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, "Deep learning models for multilingual hate speech detection," 2020, *arXiv:2004.06465*. [Online]. Available: http://arxiv.org/abs/2004.06465

[26] G. I. Sigurbergsson and L. Derczynski, "Offensive language and hate speech detection for Danish," 2019, *arXiv:1908.04531*. [Online]. Available: http://arxiv.org/abs/1908.04531

[27] J. A. García-Díaz, M. Cánovas-García, R. Colomo-Palacios, and R. Valencia-García, "Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings," *Future Gener. Comput. Syst.*, vol. 114, pp. 506–518, Jan. 2021.

[28] Z. Mossie and J.-H. Wang, "Vulnerable community identification using hate speech detection on social media," *Inf. Process. Manage.*, vol. 57, no. 3, May 2020, Art. no. 102087.

[29] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 29–30.

[30] S. G. Roy, U. Narayan, T. Raha, Z. Abid, and V. Varma, "Leveraging multilingual transformers for hate speech detection," 2021, *arXiv:2101.03207*. [Online]. Available: http://arxiv.org/abs/2101.03207

[31] S. K. Mugambi, "Sentiment analysis for hate speech detection on social media: TF-IDF weighted N-Grams based approach," M.S. thesis, Dept. Inf. Technol., Strathmore Univ., Nairobi, Kenya, 2017.

[32] F. Poletto, M. Stranisci, M. Sanguinetti, V. Patti, and C. Bosco, "Hate speech annotation: Analysis of an Italian Twitter corpus," in *Proc. 4th Italian Conf. Comput. Linguistics, (CLiC-it)*, 2017, pp. 1–6.

[33] D. Benikova, M. Wojatzki, and T. Zesch, "What does this imply? Examining the impact of implicitness on the perception of hate speech," in *Proc. Int. Conf. German Soc. Comput. Linguistics Lang. Technol.* Cham, Switzerland: Springer, Sep. 2017, pp. 171–179.

[34] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW) Companion*, 2017, pp. 759–760.

[35] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 11, no. 1, May 2017, pp. 1–4.

[36] T. Gröndahl, L. Pajola, M. Juuti, M. Conti, and N. Asokan, "All you need is, love "Evading hate speech detection," in *Proc. 11th ACM Workshop Artif. Intell. Secur.*, Jan. 2018, pp. 2–12.

[37] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018.

[38] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Effective hate-speech detection in Twitter data using recurrent neural networks," *Appl. Intell.*, vol. 48, no. 12, pp. 4730–4742, 2018.

[39] N. A. Setyadi, M. Nasrun, and C. Setianingsih, "Text analysis for hate speech detection using backpropagation neural network," in *Proc. Int. Conf. Control, Electron., Renew. Energy Commun. (ICCEREC)*, Dec. 2018, pp. 159–165.

[40] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A BERT-based transfer learning approach for hate speech detection in online social media," in *Proc. Int. Conf. Complex Netw. Their Appl.* Cham, Switzerland: Springer, Dec. 2019, pp. 928–940.

[41] M. Polignano, P. Basile, M. De Gemmis, and G. Semeraro, "Hate speech detection through AlBERTo Italian language understanding model," in *Proc. NL4AI@ AI* IA*, Nov. 2019, pp. 1–13.

[42] M.-A. Rizoiu, T. Wang, G. Ferraro, and H. Suominen, "Transfer learning for hate speech detection in social media," 2019, *arXiv:1906.03829*. [Online]. Available: http://arxiv.org/abs/1906.03829

[43] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in Indonesian Twitter," in *Proc. 3rd Workshop Abusive Lang. Online*, 2019, pp. 46–57.

[44] N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D.-Y. Yeung, "Multilingual and multi-aspect hate speech analysis," 2019, *arXiv:1908.11049*. [Online]. Available: http://arxiv.org/abs/1908.11049

[45] P. Kapil and A. Ekbal, "A deep neural network based multi-task learning approach to hate speech detection," *Knowl.-Based Syst.*, vol. 210, Dec. 2020, Art. no. 106458.

[46] K. Miok, B. Skrlj, D. Zaharie, and M. Robnik-Sikonja, "To BAN or not to BAN: Bayesian attention networks for reliable hate speech detection," 2020, *arXiv:2007.05304*. [Online]. Available: http://arxiv.org/abs/2007.05304

[47] U. Naseem, I. Razzak, and P. W. Eklund, "A survey of pre-processing techniques to improve short-text quality: A case study on hate speech detection on Twitter," *Multimedia Tools Appl.*, pp. 1–28, Nov. 2020, doi: 10.1007/s11042-020-10082-6.

[48] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, "Bidirectional long short term memory method and Word2vec extraction approach for hate speech detection," *IJCCS Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 2, 2020, pp. 169–178.

[49] P. K. Roy, A. K. Tripathy, T. K. Das, and X.-Z. Gao, ''A framework for hate speech detection using deep convolutional neural network,'' *IEEE Access*, vol. 8, pp. 204951–204962, 2020.

[50] R. Cao and R. K.-W. Lee, ''HateGAN: Adversarial generative-based data augmentation for hate speech detection,'' in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6327–6338.

[51] P. Maheshappa, B. Mathew, and P. Saha, ''Using knowledge graphs to improve hate speech detection,'' in *Proc. 8th ACM IKDD CODS 26th (COMAD)*, Jan. 2021, p. 430.

[52] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, ''Comparing pre-trained language models for Spanish hate speech detection,'' *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114120.

[53] H. M. Dubey, M. Pandit, and B. K. Panigrahi, ''An overview and comparative analysis of recent bio-inspired optimization techniques for wind integrated multi-objective power dispatch,'' *Swarm Evol. Comput.*, vol. 38, pp. 12–34, Feb. 2018, doi: 10.1016/j.swevo.2017.07.012.

[54] B. Alatas, ''Sports inspired computational intelligence algorithms for global optimization,'' *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 1579–1627, Oct. 2019, doi: 10.1007/s10462-017-9587-x.

[55] R. K. Dewangan, A. Shukla, and W. W. Godfrey, ''A solution for priority-based multi-robot path planning problem with obstacles using ant lion optimization,'' *Modern Phys. Lett. B*, vol. 34, no. 13, May 2020, Art. no. 2050137.

[56] M. Shehab, L. Abualigah, H. Al Hamad, H. Alabool, M. Alshinwan, and A. M. Khasawneh, ''Moth-flame optimization algorithm: Variants and applications,'' *Neural Comput. Appl.*, vol. 32, no. 14, 2020, pp. 9859–9884.

[57] R. S. Selvan and K. Arutchelvan, ''An efficient social spider optimization algorithm based multi-document summarization model,'' in *Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT)*, Jan. 2021, pp. 855–860.

[58] A. Hosseinalipour, F. S. Gharehchopogh, M. Masdari, and A. Khademi, ''Toward text psychology analysis using social spider optimization algorithm,'' *Concurrency Comput., Pract. Exper.*, 2021, Art. no. e6325, doi: 10.1002/cpe.6325.

[59] D. Singh, ''A new bio-inspired social spider algorithm,'' *Int. J. Appl. Metaheuristic Comput.*, vol. 12, no. 1, pp. 79–93, Jan. 2021.

[60] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, ''Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization,'' *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541, doi: 10.1016/j.engappai.2020.103541.

[61] E. H. Houssein, B. E. D. Helmy, A. A. Elngar, D. S. Abdelminaam, and H. Shaban, ''An improved tunicate swarm algorithm for global optimization and image segmentation,'' *IEEE Access*, 9, pp. 56066–56092, 2021, doi: 10.1109/ACCESS.2021.3072336.

[62] L.-L. Li, Z.-F. Liu, M.-L. Tseng, S.-J. Zheng, and M. K. Lim, ''Improved tunicate swarm algorithm: Solving the dynamic economic emission dispatch problems,'' *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107504, doi: 10.1016/j.asoc.2021.107504.

[63] T. Fetouh and A. M. Elsayed, ''Optimal control and operation of fully automated distribution networks using improved tunicate swarm intelligent algorithm,'' *IEEE Access*, vol. 8, pp. 129689–129708, 2020, doi: 10.1109/ACCESS.2020.3009113.

[64] S. Mirjalili, ''The ant lion optimizer,'' *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[65] S. Mirjalili, ''Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,'' *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.

[66] Kaggle.com. (2015). *Detecting Insults in Social Commentary*. Accessed: Jan. 1, 2021. [Online]. Available: https://www.kaggle.com/c/detecting-insults-in-social-commentary/overview

[67] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, ''SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter,'' in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 54–63.

**CEM BAYDOGAN** was born in Elâziğ. He received the B.Sc. and M.Sc. degrees in software engineering from Firat University, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree in software engineering. He is currently a Research Assistant in software engineering with Firat University. His research interests include social network analysis and meta-heuristic optimization.

**BILAL ALATAS** received the B.S., M.S., and Ph.D. degrees from Firat University, Elâziğ, Turkey. He is currently the Head of the Department of Software Engineering, Firat University, where he works as a Professor of software engineering. He is also the Founder Head of the Department of Computer Engineering, Munzur University, and the Department of Software Engineering, Firat University. He has published over 100 papers in many well-known international journals and proceedings of the refereed conference, since 2001. His research interests include artificial intelligence, data mining, social network analysis, meta-heuristic optimization, and machine learning. He has been an editor of 12 journals, four of which are indexed in SCI, and a reviewer of 60 SCI-indexed journals.