Abd El-Mageed *et al. Journal of Big Data* (2024) 11:41
https://doi.org/10.1186/s40537-024-00895-9

Journal of Big Data

RESEARCH Open Access

An adaptive hybrid african vultures-aquila optimizer with Xgb-Tree algorithm for fake news detection

Amr A. Abd El-Mageed[1], Amr A. Abohany[2], Asmaa H. Ali[3] and Khalid M. Hosny[4*]

*Correspondence: k_hosny@zu.edu.eg

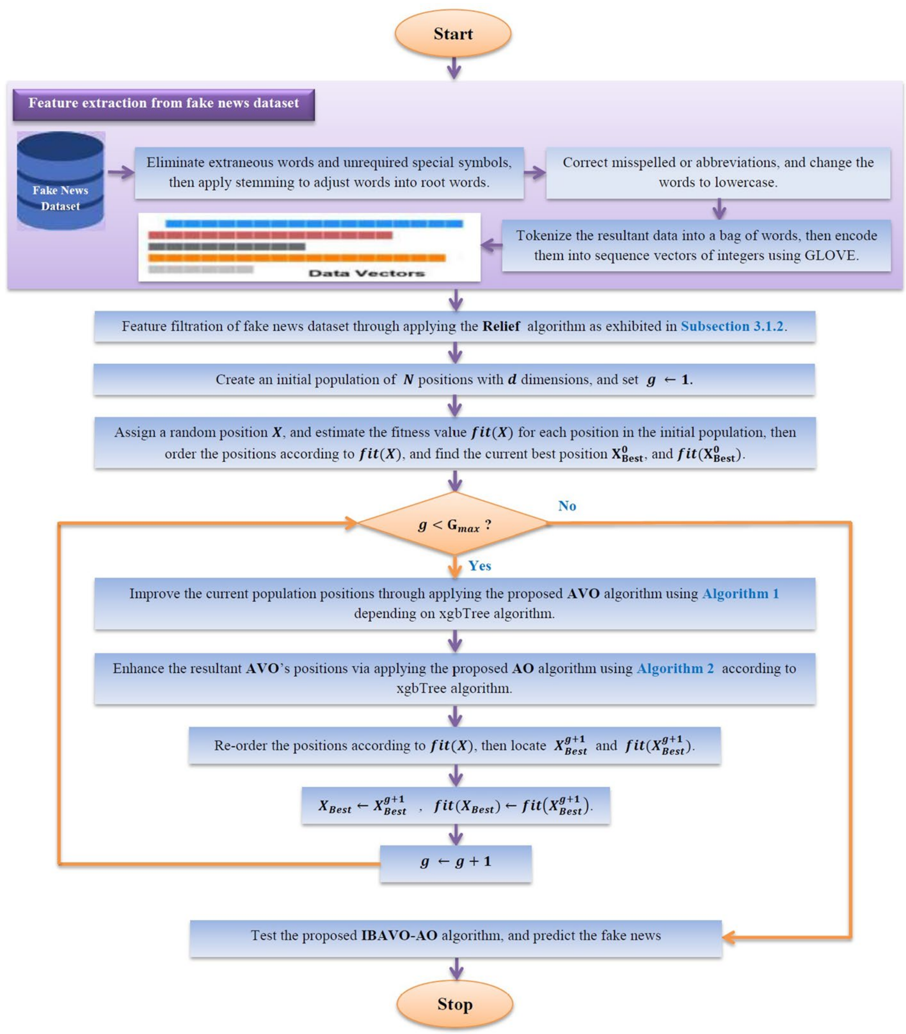
[1] Department of Information Systems, Sohag University, Sohag 82511, Egypt
[2] Department of Information Systems, Faculty of Computers and Information, Kafrelsheikh University, Kafrelsheikh 33516, Egypt
[3] Department of Information Systems, Faculty of Computers and Information, Zagazig University, Zagazig 44519, Egypt
[4] Department of Information Technology, Faculty of Computers and Information, Zagazig University, Zagazig 44519, Egypt

Abstract

Online platforms and social networking have increased in the contemporary years. They are now a major news source worldwide, leading to the online proliferation of Fake News (FNs). These FNs are alarming because they fundamentally reshape public opinion, which may cause customers to leave these online platforms, threatening the reputations of several organizations and industries. This rapid dissemination of FNs makes it imperative for automated systems to detect them, encouraging many researchers to propose various systems to classify news articles and detect FNs automatically. In this paper, a Fake News Detection (FND) methodology is presented based on an effective IBAVO-AO algorithm, which stands for hybridization of African Vultures Optimization (AVO) and Aquila Optimization (AO) algorithms, with an extreme gradient boosting Tree (Xgb-Tree) classifier. The suggested methodology involves three main phases: Initially, the unstructured FNs dataset is analyzed, and the essential features are extracted by tokenizing, encoding, and padding the input news words into a sequence of integers utilizing the GLOVE approach. Then, the extracted features are filtered using the effective Relief algorithm to select only the appropriate ones. Finally, the recovered features are used to classify the news items using the suggested IBAVO-AO algorithm based on the Xgb-Tree classifier. Hence, the suggested methodology is distinguished from prior models in that it performs automatic data pre-processing, optimization, and classification tasks. The proposed methodology is carried out on the ISOT-FNs dataset, containing more than 44 thousand multiple news articles divided into truthful and fake. We validated the proposed methodology's reliability by examining numerous evaluation metrics involving accuracy, fitness values, the number of selected features, Kappa, Precision, Recall, F1-score, Specificity, Sensitivity, ROC_AUC, and MCC. Then, the proposed methodology is compared against the most common meta-heuristic optimization algorithms utilizing the ISOT-FNs. The experimental results reveal that the suggested methodology achieved optimal classification accuracy and F1-score and successfully categorized more than 92.5% of news articles compared to its peers. This study will assist researchers in expanding their understanding of meta-heuristic optimization algorithms applications for FND.

Keywords: Fake news detection (FND), African vultures optimization (AVO), Aquila optimization (AO), Extreme gradient boosting tree (Xgb-Tree), GLOVE, Relief algorithm, Meta-heuristic

Graphical Abstract



Introduction

FND techniques have been getting extra attention since the circulation of disinformation has increased on the Internet, which has become a concern of the modern community [1]. Generally, the concept of FNs has been around for a while. This problem existed before the growth of the Internet. Many publishers utilize misinformation to promote their interests [2]. Many publishers publish FNs through convenient print media news and online platforms. Online platforms play an essential role in disseminating FNs in the community; these online platforms, such as online newspapers and social media, provide users access to various publications in one session to provide greater ease and speed than printed news media. In addition, the nature of social networks suggests an accessible platform for the fast dissemination of information in real-time; even with the reliability of this information, it has caused severe information credibility problems [3].

Not only do FNs negatively affect individuals, but they devastate the community as a whole over time. For example, FNs went viral on Facebook in the US 2016 presidential election instead of the more popular and trusted traditional news sources [4], revealing that readers may pay more attention to FNs than truthful news. Social media users who participate in spreading disinformation can have many motivations for spreading such information online, such as manipulation, political agendas, and influence. Still, while many of these users are genuine, those spreading disinformation may or may not be genuine users [5]. Because social media profiles are inexpensive and uncomplicated, many people have created social media profiles for malicious tasks. If a computer algorithm manages social media profiles, it will be used as a social bot [4]. These social bots can interact with individuals via social media and automatically produce and publish content online, making it significantly challenging for individuals to recognize such manipulated content [6].

Therefore, it isn't easy to validate online content using manual methods because, in recent years, a large amount of online content has been created and published online. Moreover, many researchers emphasized that automated and computerized FND methods should be used and are necessary [7]. FND systems have generally been divided into "news content" and "social context" classes using their information sources. The first class is "news content" techniques, which attempt to validate news content and utilize attributes such as body text, title, and more metadata to recognize FNs. These techniques are called "content-driven" techniques [8]. The Second category is social context techniques that focus on social attributes such as users' interactions in social media with specific news (liking it or sharing it on Facebook, retweeting it on Twitter). These techniques are referred to as "social-driven" techniques [8].

Deep Learning (DL) and machine learning methods have been employed in different regions [9–11] and have recently been used to tackle FND issues effectively and efficiently [12]. The leading cause for effective outcomes utilizing DL techniques is the large data volume and high dimensionality of the data for FNs. Today's scenario is a fast and large-scale growth of social media, and people are using social media to view the latest updates. Thus, social media platforms such as WhatsApp, Twitter, Facebook, and YouTube struggle to detect FNs from many user posts. There is a potential danger of publishing and disseminating such FNs via social media Platforms [13]. Many challenges must be considered when working in these areas, including selecting the most appropriate attributes, high-dimensionality data, heterogeneity, and choosing the most appropriate DL technique [14].

[15] proposed a DL method based on an automated detector via a three-level hierarchical focus network for fast and accurate FND. [16] proposed deep Convolutional Neural Networks (CNNs) for detecting FNs. [14] presented a learning model based on linguistic features to detect FNs. [17] presented a method for FND using a hybrid neural network structure, integrating the power of Long Short-Term Memory (LSTM) and CNNs. [13] presented several attributes-oriented methods for the automated detection of FNs on social media employing DL. [18] presented three DL-based models intended to classify and detect FNs. [19] presented a method for FND employing

a geometric DL. [20] introduced a neural network method to accurately forecast the stance between a given pair of headlines and the text of the article. [21] introduced several methods for FND based on the relationship between the headlines and the body of the articles. Their methods are primarily based on Bidirectional-LSTM, CNN, and LSTM.

Due to their effective performance in addressing many optimization problems, meta-heuristic algorithms have attracted much attention recently. Therefore, MHA is an efficient solution-finding method to detect FNs on social media. [22] introduced the issue of detecting FNs as an optimization problem. This study proposes two meta-heuristic algorithms, Grey Wolf Optimization (GWO) and Salp Swarm Optimization (SSO), for tackling the FND issue. The proposed FND approach is initialized through a pre-processing phase and then utilizes GWO and SSO to handle the FND issue. The suggested approach was verified utilizing three real-world FNs datasets. The experimental outcomes show that the GWO optimization algorithm achieved optimal results in different performance metrics than the SSO optimization algorithm and other meta-heuristic algorithms. [23] improved their study by proposing a new method that integrated MHA and text mining to discover FNs via online social media. Modified variants of GWO and SWO optimization algorithms based on nonlinear decreasing coefficient and oscillating inertia weight are used for the FND issue. The evaluation measures of the suggested approaches are verified on different datasets. The empirical outcomes revealed that the proposed new approaches exceeded other approaches in real-world FNs datasets. [24] introduced a new method for identifying FNs articles using the WOA-Xgb-Tree technique and content-driven attributes. The suggested model can be implemented in several scenarios for classifying news articles. The proposed model has two phases: first, the necessary attributes are identified and investigated. Then, the Xgb-Tree optimizer tuned by the Whale Optimization Algorithm (WOA) classifies the news articles using the specified attributes. In their empirical results, They considered F1-score and classification accuracy as the basis of their investigations. Then, they compared the results of their proposed system to various modern classification techniques using a dataset that has collected more than 40,000 news articles recently. The empirical outcomes reveal that the suggested system obtained a reliable F1-score rate and efficiently classified more than 91 percent of the articles.

Motivations

This paper presents a framework relying on the IBAVO-AO algorithm to tackle the issue of FND. The proposed IBAVO-AO is a hybrid AVO-AO optimizer with an Xgb-Tree classifier. The primary stages of the suggested methodology are as follows: Firstly, the collected unstructured data is converted into structured data for usage in the classification process, known as data pre-processing. In this stage, beneficial features are extracted by removing superfluous words and unnecessary special symbols, stemming from altering words into root words, tokenizing the resulting data into a bag of words, and finally encoding and padding words into sequence vectors of numerical values using Global Vectors (GLOVE) [25, 26], which is a count-based approach for pre-training and relies on terms or vectors from co-occurrence data. After that, the extracted features are filtered using an efficient Relief algorithm to determine only the associated features and provide the final classification dataset. Using the Relief algorithm aims to enhance the ability to explore the best outcomes discovered inside the solution space. In the final stage, the classification process utilizes the IBAVO-AO algorithm based on the Xgb-Tree classifier with high detection performance. The effectiveness of the suggested methodology is assessed by employing a variety of evaluation metrics and applying them to the ISOT-FNs dataset that includes more than 44 thousand news articles. After the suggested methodology has been evaluated and compared with state-of-the-art optimization techniques [27, 28], the results indicate that the presented methodology produces high classification accuracy. It is advised to use it in the FND problem.

Contributions

This paper offers an FND methodology based on the IBAVO-AO algorithm with Xgb-Tree classifier; its fundamental contributions can be clarified in the following points:

- Pre-process the FNs data to extract the necessary features.
- For improving and reducing the initial search space exploration capacity and enhancing the acquired optimal outcomes, the proposed IBAVO-AO algorithm embeds a Relief algorithm with the hybridization of AVO and AO algorithms. This embedding enhances the algorithm's performance by producing a new population that maintains the fundamental structure but has more appropriate positions.
- Filter and determine only the most appropriate features for predictive modeling using the Relief algorithm.
- Classify the news items utilizing the IBAVO-AO with the Xgb-Tree classifier.
- Assess the proposed methodology against state-of-the-art optimization algorithms using a variety of evaluation metrics involving accuracy, fitness values, the number of selected features, Kappa, Precision, Recall, F1-score, Specificity, Sensitivity, ROC_AUC, and MCC toward the ISOT-FNs dataset.
- The proposed methodology outcomes achieve high classification accuracy and a positive impact compared to its peers.

Structure

The remainder of the paper is formulated as follows. Section "Literature review" presents the related work and literature review. The proposed methodology and its components for FND are presented in Sect. "The proposed IBAVO-AO algorithm for FND". Section "Experimental results and analysis" shows the numerical results and comparisons. Finally, conclusions and future work are drawn in Sect. "Conclusion and future work".

Literature review

The primary purpose of fact-checking is to use new technologies to recognize unreliable and manipulated news content on the Internet. It is an attractive major topic within specific streams of information and library science [29]. As a result, many researchers are trying to address the issues of FNs in different areas, especially online news. This section will survey the various methods utilized to discover FNs on online platforms, and we will briefly mention their results and advantages.

[30] has proposed a new model for detecting real and fake stories. They used linguistic attributes like special characters, emoticon symbols, negative/positive words, and hashtags to categorize news stories. [31] suggests a system for detecting information sequences in a Twitter OP. Within their work, patterns analysis methods were implemented, allowing them to classify original and FNs. [32] proposed a graph kernel-based Support vector Machine (SVM) method that learns high-order distribution patterns to detect FNs. [33] proposed a novel model that uses a Recurrent Neural Network (RNN) to identify FN articles utilizing linguistic attributes accessed from a collection of user comments.

[34] introduced a novel system to identify authentic news articles. They utilized inevitable connections among conversation parts to identify trustworthy news stories. In another system, [35] analyzed the same user features on the social platform Sina Weibo, the most popular Chinese microblogging site. [36] proposed an approach to identify sarcastic tweets and product reviews automatically. They have used generic attributes based on baseline and lexicon features. The features such as character n-grams, word n-grams, and word skip-grams are extracted and integrated with lexicon properties. Then, they categorized these features utilizing different methods, such as ensemble classifiers, Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF).

[37] studied the value of various attributes in identifying and categorizing sarcastic and ironic reviews on different types of products. Firstly, they elicited attributes utilizing lexicon-based features and Bag-of-Words. Then, they used these elicited attributes on various Machine Learning (ML) classifiers such as LR, SVM, DT, and RF. [38] suggested a novel system to recognize truthful news. They integrated the different user, linguistic, structural, and temporal features to categorize FNs.

[39] proposed a method to address the problem of FNs based on DL. The suggested method contains three stages: text encoding, feature extraction, and classification. The text encoding phase is performed on the entered news words utilizing GLOVE to represent the words. The encoded words at a given word length are then included to be enrolled in the suggested DL methods. The suggested DL methods include both automated feature extraction and classification capabilities. Moreover, this search presents four DL methods containing CNNs and Concatenated CNNs, Gated Recurrent Units, and LSTM to obtain an optimum method before the problem of FNs exceeds previous studies. The suggested DL methods are implemented on FNC and FNs datasets supported by Kaggle. The proposed Concatenated CNNs method achieved a classification accuracy of 99.6% and trained faster than others.

Table 1 presents the state-of-the-art papers implemented in FND, including the dataset, model description, limitations, advantages, and outcomes. According to the outcomes of Table 1, there are many issues still open in this area, which can be summarized in the following points:

- The limited suitable quality-labeled benchmark datasets.
- Little studies have been implemented on regional languages.
- Lack of a comprehensive standardized data set on FNs.

Table 1 Comparison between various methods to identify FNs

Authors	Dataset utilized	Methodology	Limitations	Advantages	Outcomes
[12]	ISOT and FA-KES	The study introduced a novel hybrid DL approach integrating RNNs and CNNs for FND	The approach requires combination with task-specific feature engineering methods to be useful	The proposed model introduced the concept of generalized methods in handling FND	The suggested approach exceeds other non-hybrid approached in terms of classification accuracy
[13]	More than 15 thousands of news from different users of the Facebook containing both truthful news and FNs	Incorporating the behaviour of various attributes linked to Facebook accounts and examining those accounts behaviour utilizing DL and ML methods.	DL methods need more time to train and test than MI methods	Check the features of both user content and news content to identify FNs on Facebook	Accuracy LR = 99.0, DT = 99.1, *k*-NN = 99.3, SVM = 99.3, LSTM = 99.4
[40]	FNs data collected by Jruvika and FNs data collected by Guilherme Pontes	A novel CNNs semi- supervised model depend on the self-ensembling method to utilize the benefit of the linguistic information and stylometric of annotated news articles and identify the hidden patterns in unlabelled data	The proposed methodology didn't utilize transformers to enhance the accuracy of classification	ConvNet filters are employed separately on headline part and body of the news articles and then chosen feature vectors were integrated to take benefit of both the slices	The proposed approach obtained 97.45% accuracy of classification on FNs data collected by Jruvika dataset
[26]	LIAR	The study presented an efficient DL approach that utilized to determine the fakeness level in the reports of news	The accuracy of classification is not effective	The proposed algorithm integrated the utilize of attention mechanisms with relevant metadata available and uses contextual embedding as word embedding	The proposed approach yielded 46.36% accuracy of classification on the LIAR dataset, which exceeded others by 1.49%
[41]	Collected 1356 records of news from multiple users via media sources such as Twitter and PolitiFact and create different datasets for the FNs and truthful news	They combined bidirectional-LSTM with CNNs networks with attention mechanism to generate more efficient accuracy after transcribing text into vectors using 100-dimensional GLOVE word embedding	The proposed approach didn't result in a more reliable classification accuracy	The proposed methodology utilized two multiple techniques	Bidirectional-LSTM with CNNs integrated network with recognition technique achieved the highest classification accuracy of 88.78%
[42]	Two real-world datasets in two languages (English and Korean)	A novel embedding technique named link2vec extended from word2vec	The performance of the suggested link2vec should be checked via various search engines such as Google	The proposed link2vec performance is greater than that of text based methods in the two language FNs datasets	The proposed methodology outperformed text based techniques in the two utilized datasets
[43]	Aggregate real datasets from Twitter	They compared different types of ML and DL methods to discover FNs intweets, aiming at identifying patterns in both linguistic content and structure of tweets	They did not investigate the significance of image information	Combining conventional ML and DL techniques improved proposed model behaviour through the latter, while also obtaining insight towards tweets structure from the interpretability of the former	DL techniques outperformed the conventional methods, getting 99% F1 score
[16]	The dataset relates to FNs prevalence during the United States presidential election in 2016	The proposed methodology is introduced to automatically recognize the discriminating properties to classify FNs through multiple hidden layers constructed in the DNN	They didn't utilize transformers that can increase the accuracy of classification	The proposed approach will assist researchers to extend understand of the CNN-based deep models applications to detect FNs	Obtained classification accuracy of 98.36%

Table 1 (continued)

Authors	Dataset utilized	Methodology	Limitations	Advantages	Outcomes
[44]	Buzzfeed corpus, SFL dataset, FND dataset, and satire political dataset	Introduced ensemble learning framework combining four different methods called embedding depth LSTM , LSTM, LIWC CNN, and N-gram CNN for detecting FNs.	They need to investigate grammatical analysis in depth	Create optimized weights , improve the precision rate, and investigate the intractability problem through different domains	They achieved classification accuracy of 99.4%
[45]	Twitter (shared news and user profiles)	Describe a deep analysis of the characteristics which from a human and and automation views are more predictive to identify social network profiles which distributes FNs in the online environment.	The proposed model need to be verified on multiple datasets of news and users to test the accuracy of prediction	Identify the trusted profiles. Verify the reliability of social information and distributed articles.	Displaying better information enables humans and machines to determine malicious users, the average classification accuracy of 90%
[46]	COVID-19, LIAR, and ISOT	The proposed model integrated multi-layer perceptron, single layer perceptron, and CNN after the embedding layer which including pre-trained approached such as BERT, RoBERTa, GPT2, and Funnel Transformer to benefit from the deep contextual representation produced by those models and classifications ability of deep neural models	They need to use user profiles for more information	Obtaining positive outcomes without using extra attributes and network data improves the capability of learning, making contextual-dense embedding of input texts	The empirical outcomes show enhancements in the classification accuracy in comparison with traditional approaches in the same datasets
[47]	FNs or truthful news dataset, LIAR , and corpus both truthful news and FNs	Proposed the pre-trained methods for detecting FNs along with conventional and DL approaches and compare their results from different views	They need to construct health related FNs methods that will be deployed in social media during the COVID-19 pandemic	Their findings will help the research community to investigate more and news sites to select the most appropriate approach for detecting FNs	Pre-trained methods behave well to detect FNs, specially with small-scale datasets

- Very few studies have been done on detecting FNs as a multi-category classification problem.
- The DL algorithms have poor re-usability and transfer learning capability.
- The accuracy of classification is not effective in many studies.
- The performance of many of these studies in detecting FNs is still insufficient.

These issues encouraged us to introduce a novel approach to classifying news articles automatically utilizing content-based attributes and useful linguistics and the proposed hybrid algorithm based on AVO and AO with the Xgb-Tree algorithm for detecting FNs with high performance.

The proposed IBAVO-AO algorithm for FND

Data pre-processing

During this stage, the unstructured data gathered from the suggested FNs dataset (see Sect. "Dataset description") is transformed into structured data for classification. The two primary methods of this stage are to extract features from the FNs dataset, then filter the consequent features and elect only the pertinent ones. These methods are covered in the subsequent subsections.

Feature extraction

This method is carried out as follows: first, extraneous words and special symbols that are not required, such as digits, stop-words, words with only one letter, commas, hashtags, punctuation marks, etc., are eliminated from the unstructured data for an FNs dataset. Then, the necessary words are adjusted into root words, called stemming, by ignoring suffixes, affixes, inserts, and a mix of starting and ending on derived words. After correcting misspellings or abbreviations, the remaining words are changed to lowercase to handle a unified form.

The resultant data are tokenized by segregating them into a bag of words (small tokens) to obtain the words that have value in the created matrix for employment in the classification process. Additionally, the data is encoded into sequence vectors using the GLOVE method [25] for word representation, which turns tokens into a sequence of integers. Since the encoded data used are of varying lengths, each sequence vector is padded to ensure that all sequences have identical lengths; this is done by padding zero at the start of each sequence until each one equals the maximum length specified for each padded vector, which is set to 1000. Labels are also encoded, with positive labels encoded as one and negative labels encoded as zero.

Feature filtration

The sheer number of features is considered one of the significant challenges in the data pre-processing stage. Processing time and computing effort are often increased while dealing with these numerous features. Also, it could hurt classification performance. Thus, there is a need to introduce an effective method for filtering the features and picking appropriate ones. This paper suggests a straightforward and quick filtering method called the Relief algorithm [48, 49], which is suggested to identify related features.

This method focuses only on pertinent features and reduces the initial search space by locating features with comparable values for identical close samples and significant for the difference between dissimilar samples. According to the features' weighted ranking, the algorithm works as follows: First, it distinguishes between Near-Hit samples related to congruent class samples and Near-Miss samples related to mismatched class samples. The weight of the feature is then evaluated based on the Near-Hit and Near-Miss values to assess the suitability of the classification process. The features are then ranked according to their weights from the biggest to the least. The following equation can be employed to evaluate the feature weight W_A:

$$\mathcal{W}_A = \sum_{j=1}^{N} \left(X_A^j - NM(X^j)_A \right)^2 - \left(X_A^j - NH(X^j)_A \right)^2. \quad (1)$$

where $\mathcal{W}_A$ denotes the weight of the feature, N is the sample number, and X_A^j means the feature value A of data X^j. $NH(X^j)$ and $NM(X^j)$ indicate the nearest data points to X^j related to the identical and the distinct class, respectively. The Relief algorithm holds significance in feature filtration and is suitable for this problem due to the following key reasons:

- *Robustness with noisy data*: The Relief algorithm is known for its robustness in handling noisy data. If the presented datasets contain noise or outliers, Relief can perform well despite these challenges. It evaluates feature importance by considering the proximity between instances, which helps mitigate the impact of noise or irrelevant features.
- *Capability to identify relevant features*: The Relief algorithm is designed to identify the most relevant features by considering their contribution.
- *Balancing feature relevance*: The Relief algorithm considers feature relevance and redundancy factors. It helps identify a subset of features that contribute most substantially to the model's performance by differentiating between those that may be redundant and those most relevant to the target variable.
- *Bias-Free feature selection*: the Relief algorithm is less prone to exhibit bias in feature selection since it does not make any assumptions about any specific data distribution. This feature guarantees a more impartial assessment of feature significance.
- *Efficiency in handling high-dimensional data*: The Relief algorithm performs well and is appropriate for datasets with substantial features, handling high-dimensional data efficiently. Compared to certain feature selection methods, Relief could be preferred as it tends to perform well without suffering from the curse of dimensionality.
- *Handling diverse data types*: The Relief algorithm is adaptable to a wide range of dataset types since it can handle continuous and categorical features.
- *Simple implementation and interpretability*: The Relief algorithm is comparatively simple and easy to implement and comprehend, and its outcomes are frequently interpretable. Its usefulness is improved by its simplicity, particularly when the interpretability of feature selection is important.
- *Previous success or familiarity*: The choice of the Relief algorithm might also stem from its success or prevalence in similar studies or datasets. It has been successfully applied in various problem domains, including healthcare, finance, and bioinformatics.

Overall, the Relief algorithm's significance lies in its ability to efficiently and rapidly filter out unnecessary or redundant features, leading to improved model performance and interpretability even on noisy and varied datasets.

Position improvement via proposed hybrid AVO-AO optimization algorithm

The proposed AVO algorithm

An efficient nature-inspired meta-heuristic optimization algorithm termed AVO algorithm [50] is presented in this paper for modeling and imitating the natural behaviors of vultures in Africa concerning living and nutrition behavior. This algorithm is set up dependent on basic conceptions related to vultures, as follows: Initially, the AVO algorithm assumes that the population size of African vultures consists of N vultures, which vary depending on the problem being tackled. After that, the fitness function value is computed for all solutions of the African vultures' initial population, allowing the vultures' population to be tangibly split into three sets; the first set comprises the best solution, which is a vulture that is stronger than all other vultures, the second set contains the second-best solution, which is the weaker vulture than the first set, and the last set has the remaining weakest African vultures. These three sets can formulate the most significant natural function of vultures. Each set has a unique incapability to obtain and consume food. Further, the advantages and disadvantages of vultures may be reflected in the fitness function's value of the solution. As a result, the two best solutions characterize the best and strongest vultures, whereas the worst solution represents the weakest and most starved vultures. In general, the vultures attempt to retain a safe range from the worst while attempting to get close to the best vultures.

According to the conceptions mentioned above, the proposed AVO algorithm can be formulated into four essential steps to model the behavior of various vultures. These steps are depicted in the next few subsections.

Population splitting step This step aims to divide the initial population into sets by evaluating the fitness function of their solutions. The first set includes the best solution as the first set's best vulture and the second-best solution is chosen as the second set's best vulture. The residual solutions are in the final set. The population should be re-formulated for each iteration because the solutions always try to come as near as possible to the best and second-best solutions, as follows:

$$R^g = \begin{cases} BestVulture_1^g, & if\ pr^g = L_1, \\ SecondBestVulture_2^g, & if\ pr^g = L_2, \end{cases} \tag{2}$$

$$pr^g = \frac{\upsilon^g}{\sum_{g=1}^{N} \upsilon^g}. \tag{3}$$

where $BestVulture_1^g$ and $SecondBestVulture_2^g$ denote the first set's best vulture and the second-best vulture in the second set at the g^{th} iteration, respectively. pr^g is the likelihood of choosing the most suitable solution for each set at the g^{th} iteration, which is defined using the Roulette wheel procedure illustrated in Eq. (3). L_1 and L_2 are two random parameters within the range [0, 1].

Vultures' starvation level step In this step, the amount of starvation is measured for the vultures, which can also be used mathematically to model the processes of exploration, exploitation, and transformation among them. The vultures can fly farther and have more capacity to look for food when they are not starving. On the other hand, Vultures cannot fly long distances for food and might turn hostile when starving. The starvation level (F_i^g) of the i^{th} vulture at the g^{th} iteration can be expressed as follows:

$$F_i^g = (2 \times rand + 1) \times z \times \left(1 - \frac{g_i}{G_{max}}\right) + t, \tag{4}$$

$$t = h \times \left(\sin^w\left(\frac{\pi}{2} \times \frac{g_i}{G_{max}}\right) + \cos\left(\frac{\pi}{2} \times \frac{g_i}{G_{max}}\right) - 1\right). \tag{5}$$

The variable F_i^g indicates the vulture's transition from exploration to exploitation, which implies that the vultures are full. A *rand* is an arbitrary number between 0 and 1, z implies an arbitrary value within the range $[-1, 1]$, g_i signifies the present iteration's number, and G_{max} indicates the maximum iteration's number. The t value is computed by Eq. (5) to improve the effectiveness in tackling complex optimization problems and avoid falling into a local optimum. h is an arbitrary value within the range $[-2, 2]$. The predetermined constant parameter w determines the likelihood of performing the exploration process; the likelihood of exploration increases as its value rises. As its value drops, the likelihood of exploration decreases.

According to Eq. (4), the value of F_i^g progressively reduces with the increasing number of iterations. Therefore, the next step can be defined in the proposed AVO algorithm as follows:

$$\begin{cases} \textit{Exploration step}\left(\textit{vultures seek food in}\ \text{various}\ \textit{sites}\right), & \textit{if}\ \left|F_i^g\right| \geq 1, \\ \textit{Exploitation step}\left(\textit{vultures seek food in the vicinity}\right), & \textit{if}\ \left|F_i^g\right| < 1, \end{cases} \tag{6}$$

AVO's exploration step During the exploration step, the vultures are distinguished by their high capacity and optical ability to seek suitable food. Vultures are compelled to fly long distances for extended periods and inspect various random sites for food. Hence, the exploration step utilizes two distinct techniques. A predefined parameter P_1 and a random value $rand_{P_1}$ are employed to pick one of these techniques with values in the range [0, 1]. Notice that the starvation level $|F_i^g|$ in the exploration step is more major than or equal to 1. The exploration techniques can be explained as follows:

$$X_i^{g+1} = \begin{cases} R^g - D_i^g \times F_i^g, & \textit{if}\ rand_{P_1} \leq P_1, \\ R^g - F_i^g + rand \times ((UB - LB) \times rand + LB), & \textit{if}\ rand_{P_1} > P_1, \end{cases} \quad \textit{if}\ |F_i^g| \geq 1, \tag{7}$$

$$D_i^g = |(2 \times rand) \times R^g - X_i^g|. \tag{8}$$

where X_i^{g+1} is the vulture's next updated position at the next $(g+1)^{th}$ iteration, R^g is the chosen best vulture in the present iteration g, which is specified through Eq. (2), D_i^g is calculated using Eq. (8), F_i^g is the starvation level of the i^{th} vulture at the g^{th} iteration, estimated by Eq. (4). *rand* is a random value amidst zero and one; to keep food safe from other vultures and to provide a high arbitrary coefficient at the search environment scale, the vultures move randomly. *UB* indicates the variables' upper limit, *LB* presents the variables' lower limit, and X_i^g is the present position at the g^{th} iteration.

AVO's exploitation step In the AVO's exploitation step, the value of $|F_i^g|$ is smaller than 1. The exploitation step consists of two internal sub-steps, where the effectiveness of the proposed AVO algorithm is assessed. Each of these sub-steps has two distinct techniques. Two predetermined parameters with values between 0 and 1 are utilized to specify the appropriate technique in each internal sub-step: P_2 for the first sub-step and P_3 for the second sub-step. The following is an explication of these two internal sub-steps.

1. *First exploitation sub-step:* This sub-step is executed when the $|F_i^g|$ value is smaller than 1 and greater than or equal to 0.5, which utilizes two distinct techniques. A pre-defined parameter P_2 and a random value $rand_{P_2}$, with values ranging from [0, 1], are employed to decide which of these two techniques is selected.

   The first technique of this sub-step is known as *siege-fight*, in which the vultures have enough power and are moderately satiated. Because vultures gather around one specific food source, the stronger and healthful vultures attempt not to exchange food with others. In contrast, the more powerless vultures attempt to steal food from the healthful vultures by swarming close to them and starting little fights. On the other hand, the second technique is referred to as *rotational-flight*; it models and forms a spiral motion between one of the best vultures and the remaining. The techniques of the first exploitation sub-step can be illustrated as follows:

$$X_i^{g+1} = \begin{cases} D_i^g \times (F_i^g + rand) - d_t^g, & if \;\; rand_{P_2} \leq P_2, \\ R^g - (S_1^g + S_2^g), & if \;\; rand_{P_2} > P_2, \end{cases} \quad if \;\; 1 > |F_i^g| \geq 0.5, \tag{9}$$

$$d_t^g = R^g - X_i^g, \tag{10}$$

$$S_1^g = R^g \times \left(\frac{rand \times X_i^g}{2\pi} \right) \times \cos(X_i^g), \tag{11}$$

$$S_2^g = R^g \times \left(\frac{rand \times X_i^g}{2\pi} \right) \times \sin(X_i^g). \tag{12}$$

   where X_i^{g+1} denotes the vulture's next updated position at the following $(g+1)^{th}$ iteration, D_i^g is derived using Eq. (8), F_i^g indicates the degree of starvation for the i^{th} vulture at the g^{th} iteration as determined by Eq. (4), and *rand* is a random value between 0 and 1 to provide a high arbitrary coefficient. d_t^g is the distance between the vulture and one of the best two vultures, which is estimated by Eq. (10), R^g means the preferred best vulture in the present g^{th} iteration, which is set via Eq. (2), S_1^g and S_2^g

are estimated utilizing Eq. (11) and (12) respectively, and X_i^g represents the present position at the g^{th} iteration.

2. *Second exploitation sub-step:* when the value of $|F_i^g|$ is less than 0.5, this sub-step is implemented, in which numerous sieges and violent fights are performed among diverse species of vultures that have congregated around the food source. Two various techniques are used in this sub-step. To determine which of these two techniques to select, a predetermined parameter P_3 and a random value $rand_{P_3}$, with values ranging from 0 to 1, are created.

Congregate vultures around the food source is the name of the first technique of this sub-step, as diverse species of vultures are hungry so that they may attract and compete near a single food supply. Furthermore, the second technique is termed an *aggressive siege-fight*. The vultures become more offensive and attempt to scavenge the remaining food from the healthful vultures by flocking toward them in various directions. In contrast, the healthful vultures weaken and lose the power to resist the other vultures. The techniques of the second exploitation sub-step can be depicted as follows:

$$X_i^{g+1} = \begin{cases} \frac{A_1^g + A_2^g}{2}, & if\ rand_{P_3} \le P_3, \\ R^g - |d_t^g| \times F_i^g \times Levy_d, & if\ rand_{P_3} > P_3, \end{cases} \quad if\ |F_i^g| < 0.5, \tag{13}$$

$$A_1^g = BestVulture_1^g - \frac{BestVulture_1^g \times X_i^g}{BestVulture_1^g - (X_i^g)^2} \times F_i^g, \tag{14}$$

$$A_2^g = SecondBestVulture_2^g - \frac{SecondBestVulture_2^g \times X_i^g}{SecondBestVulture_2^g - (X_i^g)^2} \times F_i^g, \tag{15}$$

$$Levy_d = 0.01 \times \frac{\mu \times \sigma}{|\nu|^{\frac{1}{\beta}}}, \qquad \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\Pi\beta}{2}\right)}{\Gamma(1+\beta 2) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}}. \tag{16}$$

where X_i^{g+1} signifies the vulture's next updated position at the following $(g+1)^{th}$ iteration, which reflects the congregation of vultures. A_1^g and A_2^g are assessed by using Eq. (14) and (15) respectively, R^g means the choice best vulture at the present g^{th} iteration, which is defined via Eq. (2), d_t^g stands for the distance between the vulture and one of the best two vultures, which is estimated by Eq. (10), F_i^g indicates the degree of starvation for the i^{th} vulture at the g^{th} iteration that computed by Eq. (4), and $Levy_d$ is the function of levy flight distribution acquired by Eq. (16) to improve the efficiency of the AVO algorithm. The best vulture in the first set and the second-best vulture in the second set at the present g^{th} iteration are denoted by $BestVulture_1^g$ and $SecondBestVulture_2^g$ respectively, while the present position at the g^{th} iteration is represented by X_i^g. d is the dimensional space, μ and ν are arbitrary values evenly distributed throughout the range [0, 1], and σ is specified by Eq. (16), where $\beta = 1.5$ is a constant number.

Pseudo-code of the proposed AVO algorithm Following clarifying the critical steps of the suggested AVO algorithm illustrated above and presenting the techniques that are recommended for mimicking the natural behaviors of African vultures in living and feeding, the pseudo-code defining the proposed AVO algorithm is provided in Algorithm 1. Moreover, a flowchart of the AVO algorithm is shown in Fig. 1 to highlight its main steps.

Algorithm 1 The original AVO algorithm

Input:
N – total positions' number (size of population)
G_{max} – maximum number of permitted iterations
d – problem's dimensional space
LB – lower limit of the variables
UB – upper limit of the variables
Output:
X_{Best} – the best vulture's position located while searching
$fit(X_{Best})$ – the best fitness function value found, which should be lessened
Start
  Initialize a population of N positions, and define the required parameters' values (L_1, L_2, w, P_1, P_2, and P_3);
  Assign a random position X in the initial population;
  Estimate the fitness values $fit(X)$ for each position in the initial population;
  Rank the positions ascendingly based on their fitness function values $fit(X)$;
  $g \leftarrow 1$; ▷ present iteration number
  while $g < G_{max}$ **do**
    Set the first best vulture's position $X^g_{Best_1}$ and the second best vulture's position $X^g_{SecondBest_2}$ of the least fitness values, and their fitness values $fit(X^g_{Best_1})$ and $fit(X^g_{SecondBest_2})$ among all population's positions;
    for vulture's position $i = 1 : N$ **do**
      Depending on $X^g_{Best_1}$ and $X^g_{SecondBest_2}$, Choose the best vulture R^g,through Eq. (2);
      Upgrade the vulture's starvation level F^g_i, based on Eq. (4);
      Upgrade the levy flight distribution function $Levy_d$ using Eq. (16);
      if $|F^g_i| \geq 1$ **then**
        if $rand_{P_1} \leq P_1$ **then**
          upgrade the vulture's position X^{g+1}_i through the first exploration's condition of Eq. (7);
        else if $rand_{P_1} > P_1$ **then**
          upgrade the vulture's position X^{g+1}_i using the second exploration's condition of Eq. (7);
        end if
      else if $|F^g_i| < 1$ **then**
        if $|F^g_i| \geq 0.5$ **then**
          if $rand_{P_2} \leq P_2$ **then**
            Upgrade the vulture's position X^{g+1}_i using the first status of first exploitation sub-step by Eq. (9);
          else if $rand_{P_2} > P_2$ **then**
            Upgrade the vulture's position X^{g+1}_i using the second status of first exploitation sub-step by Eq. (9);
          end if
        else if $|F^g_i| < 0.5$ **then**
          if $rand_{P_3} \leq P_3$ **then**
            Upgrade the vulture's position X^{g+1}_i using the first case of second exploitation sub-step by Eq. (13);
          else if $rand_{P_3} > P_3$ **then**
            Upgrade the vulture's position X^{g+1}_i using the second case of second exploitation sub-step by Eq. (13);
          end if
        end if
      end if
      $fit(X^{g+1}_i) \leftarrow$ Assess the value of fitness function for X^{g+1}_i;
      if $fit(X^{g+1}_i) < fit(X^g_i)$ **then** ▷ Upgrade the previous position if the present novel position is better than it
        $X^g_i \leftarrow X^{g+1}_i$; $fit(X^g_i) \leftarrow fit(X^{g+1}_i)$;
      end if
    end for
    Re-rank the positions ascendingly based on their fitness function values $fit(X)$;
    Locate the best position X^{g+1}_{Best} and its best fitness value $fit(X^{g+1}_{Best})$ at completion of the present iteration $g + 1$;
    $X_{Best} \leftarrow X^{g+1}_{Best}$; $fit(X_{Best}) \leftarrow fit(X^{g+1}_{Best})$;
    $g \leftarrow g + 1$;
  end while
End

The proposed AO algorithm

In this subsection, the prey-hunting behavior of Aquila is simulated by introducing an efficacious meta-heuristic optimization algorithm inspired by nature, dubbed the AO algorithm [51]. Due to its bravery, agility, and speed, depending on steady feet and

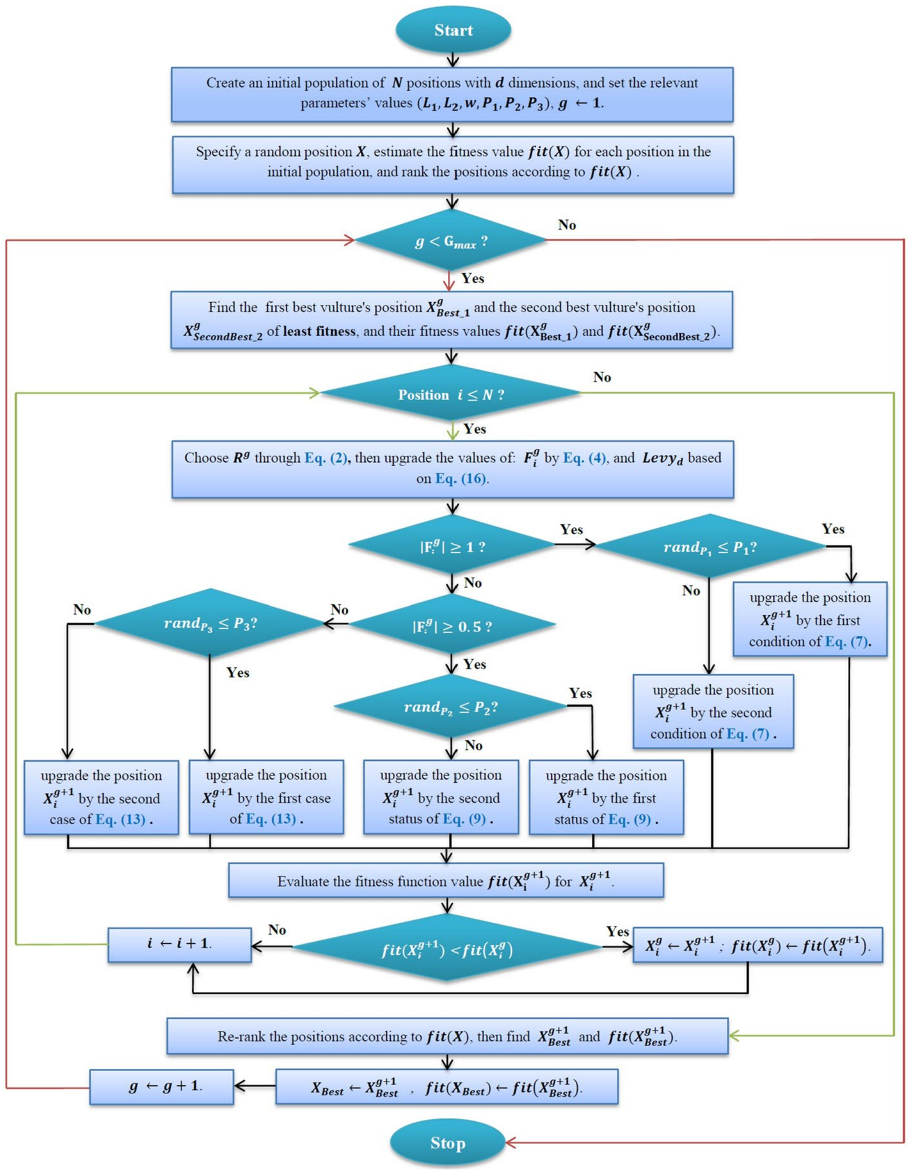


Fig. 1 Flowchart of the proposed AVO algorithm

sharp talons when hunting various animals, including badgers, squirrels, and rabbits, the Aquila is the most famous prey bird [52]. Aquila relies on four hunting techniques, which can be summed up mathematically in two crucial steps: exploration and exploitation. The appropriate step can be chosen from these two steps in the proposed AO algorithm according to the present iteration's number, g_i, and the maximum iteration's number, G_{max}, as follows:

$$\begin{cases} Explorationstep, & if \ \ g_i \leq \left(\frac{2}{3}\right) \cdot G_{max}, \\ Exploitationstep, & if \ \ g_i > \left(\frac{2}{3}\right) \cdot G_{max}. \end{cases} \tag{17}$$

The following subsections illustrate these steps of the proposed AO algorithm.

AO's exploration step The exploration step includes two distinct techniques. *Extensive exploration* is the name of the first technique, where the Aquila flies far above the land in search of suitable prey. The Aquila begins a long, low-angled glide with growing speed as it approaches the optimal region for prey. It then extends its wings and tail and lowers vertically toward the prey. On the other hand, the second technique is *restricted exploration*, in which the Aquila carefully inspects the chosen area of the prey from a high altitude, whether the prey is in flight or a running state. The Aquila then spirals around the chosen prey and goes up low off the ground to get ready to catch the prey. A random value *rand*, ranging from [0, 1], determines which of these two techniques to pick.

For improving the exploration's efficiency, the exploration step can be mathematically stated when g_i is smaller than or equal to $\left(\frac{2}{3}\right) \cdot G_{max}$. The techniques mentioned above of the AO's exploration step can be represented as follows:

$$X_i^{g+1} = \begin{cases} X_{Best}^{g} \times \left(1 - \frac{g_i}{G_{max}}\right) + (X_{Mean}^{g} - X_{Best}^{g} \cdot rand), & if \ \ rand < 0.5, \\ X_{Best}^{g} \times Levy_d + X_{\tau}^{g} + (y^g - \zeta^g) \cdot rand, & if \ \ rand \geq 0.5, \end{cases} \quad if \ \ g_i \leq \left(\frac{2}{3}\right) \cdot G_{max}, \tag{18}$$

$$X_{Mean}^{g} = \frac{\sum_{i=1}^{N} X_i^g}{N}, \ \forall j = 1, 2, ..., d, \tag{19}$$

$$y^g = (\mathfrak{r}_1 + U \times \mathfrak{D}_1) \times \cos\left(-\omega \times \mathfrak{D}_1 + \left(\frac{3 \times \Pi}{2}\right)\right), \tag{20}$$

$$\zeta^g = (\mathfrak{r}_1 + U \times \mathfrak{D}_1) \times \sin\left(-\omega \times \mathfrak{D}_1 + \left(\frac{3 \times \Pi}{2}\right)\right). \tag{21}$$

where X_i^{g+1} denotes the aquila's next updated position at the subsequent $(g+1)^{th}$ iteration, X_{Best}^{g} indicates the present best position found during searching at g^{th} iteration. g_i means the current iteration, while G_{max} means the maximum allowed iterations' number, the phrase $\left(1 - \frac{g_i}{G_{max}}\right)$ is used to dominate the extended exploration throughout the set of iterations. X_{Mean}^{g} is the mean value of the present positions at g^{th} iteration, which is evaluated through Eq. (19). The number of permitted positions is *N*, and *d* is the problem's dimension size. $Levy_d$ is the levy flight distribution function, emanated using Eq. (16). X_{τ}^{g} is a randomly chosen Aquila's position. The twisting form in the search is represented by y^g and ζ^g, which are evaluated using Eq. (20) and (21), respectively. $\mathfrak{r}_1$ indicates the number of search rotations ranges from 1 to 20, and $U = 0.00565$. $\mathfrak{D}_1$ is integer numbers from 1 to d, and $\omega = 0.005$.

AO's exploitation step Two diverse techniques are used in the exploitation step. The first technique is dubbed *extensive exploitation*. In this technique, the Aquila lands on the ground after exactly locating and exploiting the prey region and slowly approaches it for catching. This technique suits slow-moving prey or prey that lacks an escape response. *Restricted exploitation* is the name of the second technique used in the exploitation step, in which the Aquila moves on the ground as it nears and attacks its prey at the last location by following its random motions. A random number *rand*, with a value between [0, 1], is employed to choose between these two techniques.

Mathematically, in the exploitation step, when g_i is greater than $\left(\frac{2}{3}\right) \cdot G_{max}$, X_i is modified for enhancing the exploitation's performance. The aforementioned exploitation techniques of the AO can be illustrated as follows:

$$X_i^{g+1} = \begin{cases} (X_{Best}^g - X_{Mean}^g) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta, & if\ rand < 0.5, \\ QF^g \times X_{Best}^g - (\mathcal{Q}_1^g \times X_i^g \times rand) - \mathcal{Q}_2^g \times Levy_d + rand \times \mathcal{Q}_1^g, & if\ rand \geq 0.5, \end{cases} \quad if\ g_i > \left(\frac{2}{3}\right) \cdot G_{max}, \tag{22}$$

$$QF^g = g_i^{\left(\frac{2 \times rand - 1}{(1 - G_{max})^2}\right)}, \tag{23}$$

$$\mathcal{Q}_1^g = 2 \times rand - 1, \quad \mathcal{Q}_2^g = 2 \times \left(1 - \frac{g_i}{G_{max}}\right). \tag{24}$$

where X_i^{g+1} is the aquila's next updated position at the following $(g+1)^{th}$ iteration, X_{Best}^g means the current best position found during the search at g^{th} iteration. X_{Mean}^g denotes the mean value of the present positions at g^{th} iteration, and can be assessed through Eq. (19). The exploitation step's adjustment parameters, α, and δ, are given to (0.1). UB and LB indicate the upper and lower limits of the search space, respectively. The search strategy is balanced using the quality function value QF^g, which is calculated using Eq. (23). Aquila's arbitrary motions while pursuing its prey are reflected in $\mathcal{Q}_1^g$ by Eq. (24). Aquila's flying slope when tracking its prey is represented by $\mathcal{Q}_2^g$, which decreases in value from 2 to 0 and is determined using equation Eq.(24). X_i^g is the present position at the g^{th} iteration, $Levy_d$ is the function of levy flight distribution, defined using Eq. (16), g_i means the current iteration, while G_{max} represents the maximum allowed iterations' number.

Pseudo-code of the proposed AO algorithm After introducing the steps mentioned above of the AO algorithm, exploration and exploitation, and showing the four techniques suggested to imitate Aquila's hunting behavior, the pseudo-code of the proposed AO algorithm is presented in Algorithm 2. Additionally, Fig. 2 includes a flowchart of the AO algorithm to show its main steps.

Algorithm 2 The original AO algorithm

Input:
N – total positions' number (size of population)
G_{max} – maximum number of permitted iterations
d – problem's dimensional space
LB – lower limits of search space
UB – upper limits of search space
Output:
X_{Best} – the best Aquila's position found while searching
$fit(X_{Best})$ – the best fitness function value discovered, which should be decreased
Start
Initialize a population of N positions, and set the needed parameters' values ($\mathfrak{r}_1$, U, ω, α, δ);
Assign a random position X in the initial population;
Estimate the fitness values $fit(X)$ for each position in the initial population;
Arrange the positions ascendingly based on their fitness function values $fit(X)$;
Locate the current best Aquila's position X^0_{Best} of the least fitness, and its fitness value $fit(X^0_{Best})$ through all initial population's positions;
$g \leftarrow 1$; ▷ present iteration number
while $g < G_{max}$ **do**
Calculate the values of y^g and ζ^g, according to Eq. (20) and (21);
Compute the value of QF^g, based on Eq. (23);
Find the value of $\mathcal{Q}^g_1$ and $\mathcal{Q}^g_2$, utilizing Eq. (24);
for position $i = 1 : N$ **do**
Set the mean value of the current position X^g_{Mean} through Eq. (19);
Upgrade the levy flight distribution function $Levy_d$ using Eq. (16);
if $g_i \leq \left(\frac{2}{3}\right) \cdot G_{max}$ **then**
if $rand < 0.5$ **then**
upgrade the Aquila's position X^{g+1}_i through the first exploration's case of Eq. (18);
else if $rand \geq 0.5$ **then**
upgrade the Aquila's position X^{g+1}_i based on the second exploration's case of Eq. (18);
end if
else if $g_i > \left(\frac{2}{3}\right) \cdot G_{max}$ **then**
if $rand < 0.5$ **then**
upgrade the Aquila's position X^{g+1}_i by the first exploitation's condition of Eq. (22);
else if $rand \geq 0.5$ **then**
upgrade the Aquila's position X^{g+1}_i using the second exploitation's condition of Eq. (22);
end if
end if
$fit(X^{g+1}_i) \leftarrow$ Evaluate the value of fitness function for X^{g+1}_i;
if $fit(X^{g+1}_i) < fit(X^g_i)$ **then** ▷ Upgrade the previous position if the present new position is better than it
$X^g_i \leftarrow X^{g+1}_i$; $fit(X^g_i) \leftarrow fit(X^{g+1}_i)$;
end if
end for
Re-arrange the positions ascendingly based on their fitness function values $fit(X)$;
Locate the best position X^{g+1}_{Best} and its best fitness value $fit(X^{g+1}_{Best})$ at completion of the present iteration $g+1$;
$X_{Best} \leftarrow X^{g+1}_{Best}$; $fit(X_{Best}) \leftarrow fit(X^{g+1}_{Best})$;
$g \leftarrow g + 1$;
end while
End

Adjustment of bound-constraint

This paper presents a bound-constrained adjustment method for re-positioning impractical decision variables beyond the search space's scope during position improvement employing the above-mentioned meta-heuristic optimization algorithms (AVO and AO algorithms). Using the random method that adjusts decision variable values outside the permissible limits with randomly generated ones inside those limits is recommended. This method can be mathematically stated as follows:

$$X^{adjust}_{i,d} = \begin{cases} X_{i,d}, & if\ X^{LB}_d \leq X_{i,d} \leq X^{UB}_d \\ X^{LB}_d + rand(0,1) \times (X^{UB}_d - X^{LB}_d), & if\ X_{i,d} < X^{LB}_d \\ X^{LB}_d + rand(0,1) \times (X^{UB}_d - X^{LB}_d), & if\ X^{UB}_d < X_{i,d}. \end{cases} \quad (25)$$

Where $X^{adjust}_{i,d}$ represents the value of the appropriate decision variable, $X_{i,d}$ denotes the infeasible value that is beyond the variable's limits, X^{LB}_d and X^{UB}_d depict the lower and

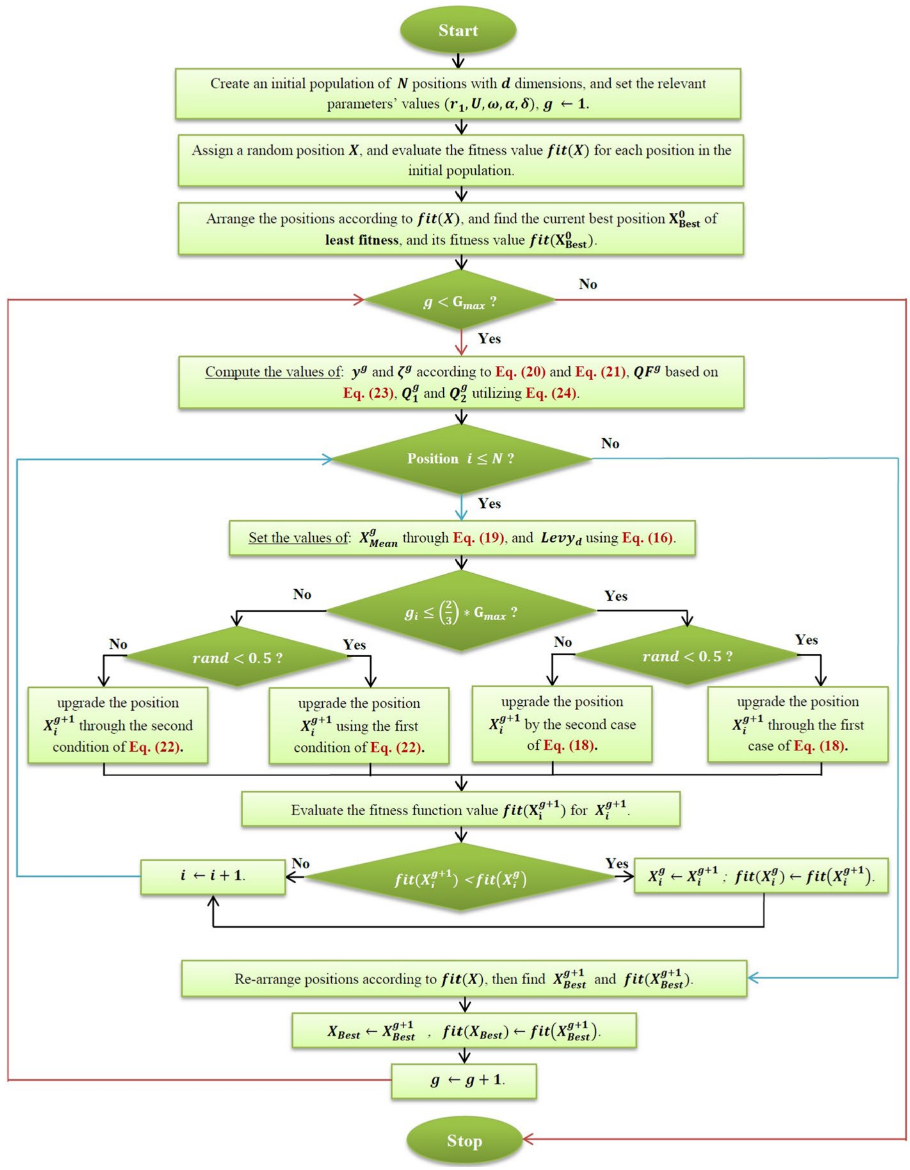


Fig. 2 Flowchart of the proposed AO algorithm

upper limits, respectively, and $rand(0, 1)$ is a random number falling within the range [0, 1].

Xgb-Tree classification algorithm

The Xgb-Tree [53] is a developed algorithm for the gradient boosting framework [54–56], which can classify sample instances into a specific class. This algorithm utilizes integrating methods and additional training procedures to integrate many weak learners to create a powerful learner. The Xgb-Tree algorithm's core concept is to boost the gradient tree by consecutively producing DTs. According to complementary models from prior iterations, boosting decreases errors and enhances classification performance. The Xgb-Tree's objective function consists of a training loss part, which is used to gauge how well the model performs on training data, and a regularization part, which handles the problems of over-fitting and model complexity. The structure score of Xgb-Tree is an objective function that is expressed as:

$$Obj = \sum_{j=1}^{T_L} \left[\mathcal{G}_j \varphi_j + \frac{1}{2} (\mathcal{H}_j + \lambda) \varphi_j^2 \right] + \gamma T_L. \tag{26}$$

Where T_L stands for the whole leaves' number on the tree, and φ_j signifies a vector value representing leaves' scores.

The proposed IBAVO-AO algorithm

Since FNs are purposefully designed to provide false information, detecting it can be challenging. This paper suggests an effective IBAVO-AO algorithm to specify FNs by combining the AVO algorithm and the AO algorithm, leading to more accurate findings. In our proposed IBAVO-AO algorithm, we tried to solve the FND problem by hybridizing the natural processes of AVO and AO. In the proposed algorithm, the AVO algorithm creates solutions in their search space and tries to improve them. After that, the AO algorithm improves the solutions produced in the space of the AVO solutions through exploration and exploitation processes. The IBAVO-AO algorithm combines the AVO and AO algorithms through the following steps:

- Firstly, within the specified search space, the population of AVO solutions with random values is initialized. the AVO algorithm handles The exploration and exploitation of the search space. The exploration step permits the algorithm to search for new areas of the search space, while the exploitation step concentrates on boosting the search around promising solutions. AVO algorithm simulates an African vulture's living and feeding behavior to improve solutions iteratively.
- The vultures navigate the search space by adjusting their positions based on their current positions and a set of candidate solutions that have already been discovered.
- Secondly, the AO algorithm is another optimization technique used in the IBAVO-AO algorithm. It operates on a set of the candidate solutions obtained by the AVO algorithm and improves them iteratively.
- The AO algorithm aims to improve the produced AVO solutions and strike a balance between exploration and exploitation capabilities. This balance facilitates the effective exploration of the search space, preventing the occurrence of local optima and enhancing the convergence towards optimum solutions.
- A set of new candidate solutions is created during the AO algorithm and combined with the population's preexisting solutions. After that, a comparison between the new candidate solutions and the original solutions is made, and chosen is performed based on the values of their objective functions.

Combining the AVO and AO algorithms involves creating a hybrid IBAVO-AO algorithm that leverages the strengths of each algorithm to improve overall optimization performance, convergence speed, and solution quality. The main advantages offered by the hybrid IBAVO-AO algorithm integrating the AVO and AO algorithms over using them separately are as follows:

- *Enhanced exploration and exploitation*: AVO and AO algorithms might be supreme in different aspects of exploration and exploitation. Combining them allows the hybrid IBAVO-AO algorithm to explore a broader solution space effectively.
- *Diversity in search*: AVO and AO algorithms have different search mechanisms, enabling the hybrid IBAVO-AO algorithm to maintain a diverse population of solutions. This diversity can prevent premature convergence to suboptimal solutions.
- *Improved convergence*: Leveraging the complementary strengths of AVO and AO algorithms, the hybrid IBAVO-AO algorithm can converge faster toward better solutions than using each algorithm separately.
- *Robustness*: The hybrid IBAVO-AO algorithm enhances the robustness of the optimization process. It will be more resilient to getting stuck in local optima.

The proposed IBAVO-AO algorithm divides News items into two class labels–Fake and Truthful–meaning that the FND issue is drafted as a binary classification. The suggested IBAVO-AO algorithm's flowchart is depicted in Fig. 3. The proposed methodology for

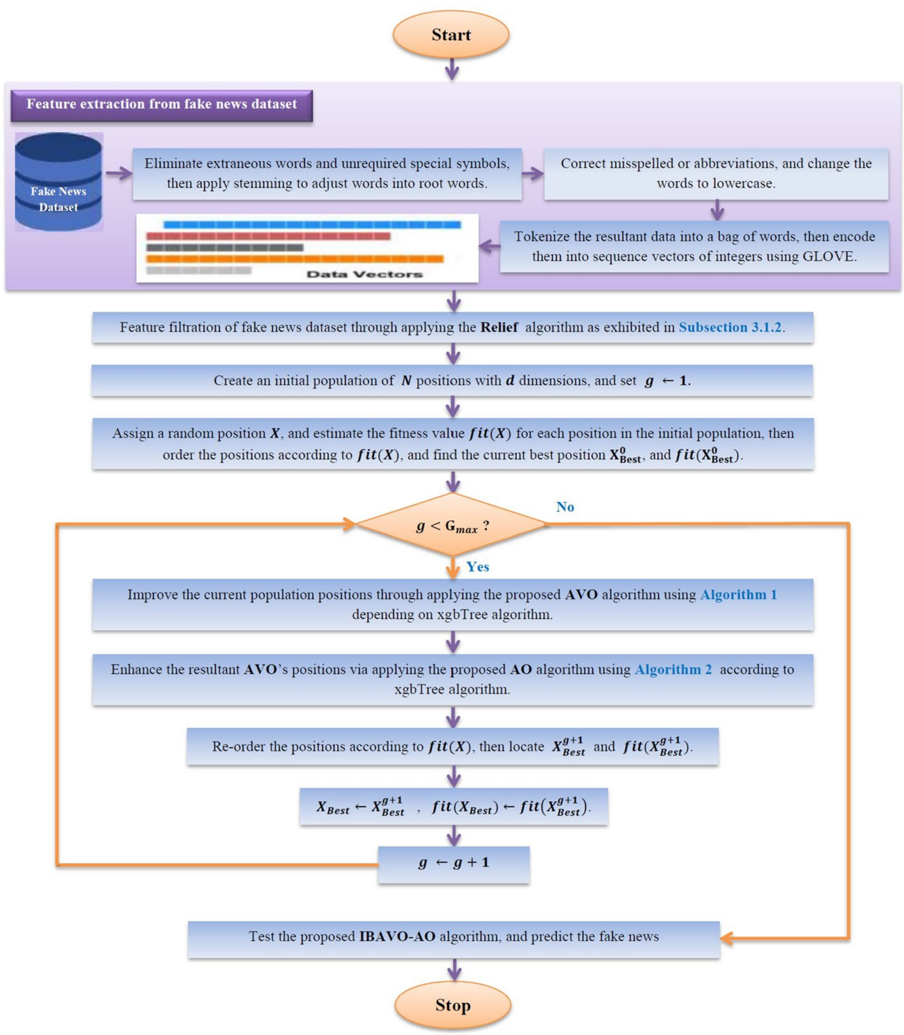


Fig. 3 Flowchart of the proposed IBAVO-AO algorithm

FND includes the following steps: Initially, as described in Sect. "Data pre-processing", the FNs dataset is pre-processed employing feature extraction and feature filtration methods. After that, the ultimate classification dataset is generated utilizing the pertinent features. Eventually, the proposed IBAVO- AVO algorithm is applied to the developed dataset, which updates the positions and determines the best values depending on the Xgb-Tree classification algorithm.

Experimental results and analysis

This section details the experimental results to evaluate the suggested FND methodology based on the IBAVO-AO algorithm with the Xgb-Tree classifier, describes the evaluation measures, and discusses the classification results.

Dataset description

The ISOT-FNs dataset [57] is an extensive collection comprising approximately 44,900 news articles. This dataset is bifurcated into two primary categories: truthful and FNs. The methodology employed in compiling this dataset is meticulous, involving a selection of news articles from various sources, each meticulously vetted for reliability. For sourcing truthful news, the dataset relies on articles from Reuters, a well-regarded international news organization known for its comprehensive and fact-based reporting. In contrast, the FNs articles are sourced from websites identified as unreliable by reputable fact-checking entities such as Politifact.com and Wikipedia.

While the ISOT-FNs dataset offers a valuable resource for studying the characteristics and spread of fake versus truthful news, it's essential to consider potential biases. The selection of sources, particularly for FNs, might reflect biases inherent in the criteria used by Politifact.com and Wikipedia. This could result in a dataset that may not fully represent the spectrum of FNs sources, especially those that are more subtle or sophisticated in their misinformation strategies.

Moreover, when comparing the ISOT-FNs dataset to other popular datasets in the field, such as the FNs Challenge (FNC-1) dataset or the Liar dataset, there are noticeable differences in size, source diversity, and categorization methodologies. For instance, the FNC-1 dataset focuses more on the stance detection between headline and body text, whereas the Liar dataset includes short statements and speeches labeled for truthfulness. These differences highlight the varying approaches in the field of FND and the importance of considering multiple datasets to gain a comprehensive understanding of the issue.

Experimental setup

FND is a complicated process, and the appropriate method requires various factors to identify manipulated news efficiently. That is the leading cause for integrating the IBAVO-AO optimization algorithm and the Xgb-Tree classification algorithm into the suggested methodology. Moreover, in contrast to the different methods presented, our suggested approach employed the Relief algorithm, which is explained in Sect. "Feature filtration", to pick only the relevant features and identify FNs articles in less time and with lower processing cost by calculating the weights of each feature in the dataset and sorting them from biggest to smallest. Lastly, the features that have small weight are removed. Upon executing the Relief algorithm on the datasets, we discovered that the greatest weights were only associated with 50 features. For this reason, just these 50 important features were selected, while the remaining irrelevant features with minor weights were omitted.

This method focuses only on pertinent features and reduces the initial search space by locating features with comparable values for identical close samples and significant for the difference between dissimilar samples.

Thus, to adequately assess the performance of the proposed system, two sets of experiments were carried out on the utilized ISOT- FNs dataset. In the first part of the experiments, we conducted a detailed comparative analysis using the ISOT-FNs dataset. To provide a robust benchmarking framework, we selected a variety of well-established classification algorithms, each known for its unique strengths in the domain of FND. These include:

- Decision Tree (DT) [58]: A simple yet powerful algorithm valued for its interpretability and ease of use in various classification problems.
- K-nearest Neighbors (k-NN) [59]: This algorithm is effective in handling multi-class classification tasks and is known for its simplicity and efficacy.
- Gaussian Naive Bayes (GNB) [60]: Chosen for its proficiency in managing high-dimensional data, GNB applies a probabilistic approach to classification.
- Support Vector Machine (SVM) [61]: Renowned for its robustness, especially in high-dimensional spaces, making it suitable for complex classification tasks.
- Random Forest (RF) [62]: Selected for its high accuracy and efficiency, especially in large datasets, RF is a versatile and powerful ensemble method.
- Multilayer Perceptron (MLP) [63]: A feedforward artificial neural network known for its ability to learn non-linear models and patterns in data. These algorithms were rigorously tested against our IBAVO-AO with Xgb-Tree classification on the same dataset, providing a comprehensive and balanced benchmarking environment.

Table 2 shows the significant parameters of the classification algorithms introduced in this paper.

Secondly, a comprehensive comparative analysis between our proposed IBAVO-AO combined with the Xgb-Tree classification algorithms and a range of widely recognized meta-heuristic optimization algorithms. These algorithms were meticulously chosen for their relevance and popularity in optimization tasks. They include:

- Binary African Vulture Optimization (BAVO) [50]: An optimization algorithm inspired by the foraging behavior of vultures, known for its efficiency in binary search spaces.
- Binary Aquila Optimizer (BAO) [51]: This algorithm mimics the hunting strategy of Aquila eagles and is notable for its precision and speed.
- Binary Sparrow Search Algorithm (BSSA) [64]: A novel algorithm based on the social behavior of sparrows, appreciated for its ability to explore and exploit the solution space.

Table 2 The major parameters of the classification algorithms

Classification algorithm	Parameters
Xgb-Tree	Number of boosting iterations $nrounds = 100$
	Maximum depth of a tree $max\_depth = 3$
	Minimum loss reduction $gamma = 0$
	Minimum sum of instance weight $min\_child\_weight = 1$
	Step size shrinkage (learning rate) $eta = 0.4$
	Sub-sample ratio of columns $colsample\_bytree = 0.8$
	Sub-sample ratio of training $sub\_sample = 0.75$
DT	Maximum depth of a tree $max\_depth = 5$
	Number of features $max\_features = 1$
k-NN	Euclidean distance metric $k = 5$
SVM	Regularization parameter $C = 1$
	Degree of polynomial kernel $degree = 2$
RF	Number of trees in a forest $n\_estimators = 10$
	Maximum depth of a tree $max\_depth = 5$
	Number of features $max\_features = 1$
MLP	Number of neurons in the i^{th} hidden layer $hidden\_layer\_sizes = (1000, 500, 100)$
	Strength of the L2 regularization term $alpha = 0.001$
	Maximum number of iterations $max\_iter = 1000$

- Binary Atom Search Optimization (BASO) [65]: Inspired by the laws of physics and molecular movement, known for its robustness in binary optimization problems.
- Binary Henry Gas Solubility Optimization (BHGSO) [66]: This algorithm simulates the gas solubility process and is recognized for its adaptability in various optimization contexts.
- Binary Harris Hawks Optimization (BHHO) [67]: Mimics the cooperative hunting strategy of Harris hawks, known for its effectiveness in complex optimization scenarios.
- Binary Sailfish Optimizer (BSFO) [68]: Based on the predatory behavior of sailfish, this algorithm is praised for its swift convergence and flexibility.
- Binary Bat Algorithm (BBA) [69]: Utilizes echolocation behavior of bats and is popular for its balance between exploration and exploitation.
- Binary Grasshopper Optimization Algorithm (BGOA) [70]: Inspired by the swarming behavior of grasshoppers, it's efficient in finding global optima in complex landscapes.
- Binary Artificial Bee Colony (BABC [71]): Mimics the foraging behavior of honey bees, well-regarded for its simplicity and effectiveness in binary domains.
- Binary Particle Swarm Optimization (BPSO) [72]: Based on the social behavior of bird flocking, this algorithm is known for its efficiency and easy implementation.

These selected algorithms represent a diverse range of strategies in meta-heuristic optimization, ensuring a robust and comprehensive benchmarking against our proposed IBAVO-AO algorithm. The comparative study was conducted on the ISOT-FNs dataset, and the specific parameters employed for each algorithm in the comparison are detailed in Table 3.

Python was used on a computing environment with a Dual Intel® Xeon® Gold 5115 2.4 GHz CPU and 128 GB of RAM on the Microsoft Windows Server 2019 operating system to run all experiments in this study. For a reliable comparison, the size of the population is estimated to be ten, and the maximum number of iterations is estimated to be 100 for all methods. Accordingly, the population size was set to 10, and the number of iterations was set to 100. Also, in this study, the new dataset is split into learning and testing after defining content-oriented attributes and creating a new dataset. Thus, 80%

Table 3 Parameter settings for used meta-heuristic optimization algorithms

Optimization algorithm	Parameter
All algorithms	Run's number = 30
	Iterations number $G_{max} = 100$
	Size of population $N = 10$
	Dimensionality d = The number of attributes in the used benchmark
BAVO	Parameter $L_1 = 0.7$
	Parameter $L_2 = 0.2$
	Parameter $w = 2$
	Parameter $P_1 = 0.6$
	Parameter $P_2 = 0.6$
	Parameter $P_3 = 0.5$
BAO	Number of search rotations $\mathfrak{r}_1 = 10$
	$U = 0.00565$
	$\omega = 0.005$
	Adjustment parameters for exploitation stage $\alpha = 0.1$ and $\delta = 0.1$
	Aquila's arbitrary motions $\mathcal{Q}_1 \in [-1, 1]$
	Aquila's flying slope $\mathcal{Q}_2 \in [2, 0]$
BSSA	Number of scroungers $SD = 0.1*N$
	Number of producers $PD = 0.2*N$
	Safety threshold $ST = 0.8$
BASO	Multiplier weight $\beta = 0.2$
	Depth weight $\alpha = 50$
BHGSO	Number of clusters = 2
	$l_1 = 5E-03, l_2 = 1E+02$, and $l_3 = 1E-02$
	$\alpha = \beta = 0.1$ and $K = 1$
BHHO	Rabbit energy $E \in [-1, 1]$
BSFO	Ratio between sardines and sailfish $pp = 0.1$
	$\varepsilon = 0.0001$
	$A = 1$
BBA	Loudness $A = 0.8$
	Lower and upper pulse frequencies = 0, 10
	Pulse emission rate $r = 0.95$
BGOA	$C_{\min} = 0.00004$ and $C_{\max} = 1$
BABC	Number of employed bees = 16
	Number of scout bees = 3
	Number of onlooker bees = 4
BPSO	Inertia weight $\omega_{max} = 0.9, \omega_{min} = 0.4$
	Acceleration coefficients ($c_2 = c_1 = 1.2$)

of the data was utilized for learning, while 20% was utilized for evaluating the proposed system. Finally, a 10-fold cross-validation method is employed to reduce model error for learning and testing purposes.

Evaluation measures

In this study, the effectiveness of the suggested IBAVO-AO with the Xgb-Tree methodology must be assessed utilizing standard metrics to ensure that the empirical outcomes are statistically valuable. To that end, the primary evaluation metric employed was accuracy [73], which is the number of successful predictions divided by the total number of predictions.

Accuracy is expressed as in Eq. (27):

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{27}$$

Where True Positive (T_P) is the percentage of FNs that were successfully classified utilizing the proposed system, True Negative (T_N) is the percentage of truthful news that was successfully classified utilizing the proposed system, False Positive (F_P) is the percentage of truthful news classified as FNs, and the percentage of FNs items classified as truthful news is represented by False Negative (F_N).

Kappa is calculated with the following formula:

$$kappa = \frac{P_o - P_e}{1 - P_e} \tag{28}$$

P_o is the model's overall accuracy, and P_e is the agreement between the model predictions and the actual class values.

Precision [74] is expressed as in Eq. (29):

$$Precision = \frac{T_P}{T_P + F_P} \tag{29}$$

Recall [75] is expressed as in Eq. (30):

$$Recall = \frac{T_P}{T_P + F_N} \tag{30}$$

F-Measure (F_1) [76] is obtained as in Eq. (31):

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{31}$$

Specificity [77] is expressed as in Eq. (32):

$$Specificity = \frac{T_N}{T_N + F_P} \tag{32}$$

Sensitivity [77] is expressed as in Eq. (33):

$$Sensitivity = \frac{T_P}{T_P + F_N} \tag{33}$$

The fitness measure calculates the mean fitness results achieved by running the suggested method separately for 30 runs, demonstrating the synergy between minimizing the number of features selected and reducing the error classification rate as Eq. (34). The minimum value presents the best result, which is assessed according to fitness as:

$$Fitness = \frac{1}{30}\sum_{k=1}^{30} f_{\cdot}^{k}, \tag{34}$$

where $f_{\cdot}^{k}$ is the optimum fitness result achieved in the k-th run.

Features Size measure shows the mean number of selected features by running the method separately 30 runs and is defined as:

$$FeaturesSize = \frac{1}{30}\sum_{k=1}^{30} \frac{|d_{\cdot}^{k}|}{|D|}, \tag{35}$$

where $|d_{\cdot}^{k}|$ is the size of features chosen in the optimal solution for the k-th run, and $|D|$ represents the complete size of features in the used benchmark.

Standard Deviation (SD): Corresponding to the measures mentioned above, the final results achieved over the 30 independent runs for each algorithm on every dataset are evaluated and analyzed in terms of stability as:

$$SD = \sqrt{\frac{1}{29}\sum\nolimits_{k=1}^{30} \left(Y_{*}^{k} - \mu_Y\right)^2}, \tag{36}$$

where Y denotes the metric to be measured, Y_{*}^{k} is the value of the metric Y in the k-th run, and μ_Y is the average of the metric over the 30 independent runs.

Effect of different components of the proposed IBAVO-AO algorithm for FND

The proposed IBAVO-AO algorithm is compared to the original versions of AVO and AO algorithms to show how this hybridization improves the performance of the IBAVO-AO algorithm. Table 4 displays the results of the proposed IBAVO-AO algorithm and its component algorithms on the utilized ISOT-FNs dataset for FND, in which boldface numbers indicate the best results.

Results analysis of the proposed IBAVO-AO algorithm versus diverse state-of-the-art ML methods and metaheuristic algorithms for FND

In the first part of the analysis of the results, we compared the empirical outcomes of the proposed IBAVO-AO algorithm with some state-of-the-art ML methods on the used ISOT-FNs dataset for FND. For a reliable comparison, the suggested system and selected methods are executed on a framework with identical parameters and tested on the same ISOT-FNs dataset.

Table 5 shows the results of the proposed IBAVO-AO algorithm and other state-of-the-art Ml methods on the utilized ISOT-FNs dataset for FND, where boldface numbers indicate the best results. Table 5 shows that the proposed IBAVO-AO

Table 4 Results of the proposed IBAVO-AO algorithm and its components for FND

Metric	IBAVO-AO	BAVO	BAO
Mean (Accuracy)	**0.9275**	0.9262	0.9261
SD (Accuracy)	**0.0006**	0.0008	0.0008
Mean (Fitness)	**0.0753**	0.0773	0.0773
SD (Fitness)	**0.0006**	0.0009	0.0008
Mean (Features size)	**17.5000**	21.1600	20.5300
SD (Features size)	02.1211	02.5701	**01.9619**
Mean (Kappa)	**0.8555**	0.8528	0.8527
SD (Kappa)	**0.0011**	0.0016	0.0015
Mean (Precision)	**0.9843**	0.9834	0.9834
SD (Precision)	0.0014	**0.0013**	0.0015
Mean (Recall)	**0.8771**	0.8753	0.8752
SD (Recall)	**0.0015**	0.0017	0.0016
Mean (F1-score)	**0.9276**	0.9262	0.9262
SD (F1-score)	**0.0006**	0.0008	0.0008
Mean (Specificity)	**0.9843**	0.9834	0.9834
SD (Specificity)	**0.0015**	0.0017	0.0016
Mean (Sensitivity)	**0.8771**	0.8753	0.8752
SD (Sensitivity)	**0.0015**	0.0017	0.0016
Mean (ROC_AUC)	**0.9657**	0.9654	0.9656
SD (ROC_AUC)	**0.0009**	0.0010	0.0011
Mean (MCC)	**0.8611**	0.8587	0.8585
SD (MCC)	**0.0012**	0.0019	0.0015

Table 5 Results of the proposed IBAVO-AO algorithm and well-known ML methods for FND

Algorithm	Accuracy	Kappa	Precision	Recall	F1-score
IBAVO-AO	**0.9275**	**0.8555**	**0.9843**	**0.8771**	**0.9276**
k-NN	0.6281	0.2580	0.6642	0.6020	0.6316
SVM	0.6418	0.3064	0.8886	0.3699	0.5223
DT	0.6133	0.1940	0.5847	0.9312	0.7183
RF	0.7157	0.4154	0.6675	0.9225	0.7746
GNB	0.8196	0.6449	0.9801	0.6730	0.7980
MLP	0.7308	0.4739	0.9321	0.5302	0.6759

algorithm and ML methods are compared and assessed regarding average accuracy, Kappa, Precision, Recall, and F1 score. According to the results obtained, the overall performance of the proposed IBAVO-AO algorithm was further compared with other ML methods on the extracted attributes. The proposed IBAVO-AO algorithm succeeded in categorizing 92.75% of the news articles. After the suggested system, GNB ranked second with a classifying rate of 81.96% of news articles, but there is a gap of more than 10% between it and the suggested system. As presented in Table 5, the DT and k-NN methods generated the least values in categorizing various news articles.

In the second part of the analysis of the results, we compared the experimental results of the proposed IBAVO-AO algorithm with some known metaheuristic optimization techniques on the used data set for FND. As shown in Table 6, the proposed

Table 6 Results of the proposed IBAVO-AO algorithm and other optimization algorithms for FND

Metric	IBAVO-AO	BSSA	BASO	BHGSO	BHHO	BSFO	BBA	BGOA	BABC	BPSO
Mean (Accuracy)	**0.9275**	0.9258	0.9211	0.9231	0.9257	0.9242	0.9225	0.9253	0.9256	0.9238
SD (Accuracy)	**0.0006**	0.0007	0.0015	0.0011	0.0008	0.0009	0.0020	0.0007	0.0008	0.0009
Mean (Fitness)	**0.0753**	0.0774	0.0834	0.0817	0.0781	0.0800	0.0817	0.0786	0.0780	0.0805
SD (Fitness)	**0.0006**	0.0008	0.0018	0.0010	0.0009	0.0008	0.0022	0.0007	0.0007	0.0010
Mean (Features Size)	**17.5000**	19.9000	26.4300	27.8300	22.7300	24.9300	24.7600	23.0000	21.8300	25.4000
SD (Features Size)	02.1211	02.5701	**01.9619**	03.0885	02.6068	03.0652	03.1164	02.8166	03.1420	02.9507
Mean (Kappa)	**0.8555**	0.8521	0.8427	0.8466	0.8518	0.8489	0.8456	0.8511	0.8518	0.8481
SD (Kappa)	**0.0011**	0.0013	0.0030	0.0021	0.0016	0.0018	0.0040	0.0015	0.0015	0.0018
Mean (Precision)	**0.9843**	0.9830	0.8599	0.9810	0.9827	0.9819	0.9812	0.9827	0.9833	0.9820
SD (Precision)	0.0014	**0.0013**	0.1442	0.0017	0.0016	0.0015	0.0020	0.0015	0.0019	0.0018
Mean (Recall)	**0.8771**	0.8751	0.8689	0.8716	0.8750	0.8729	0.8704	0.8743	0.8744	0.8720
SD (Recall)	**0.0015**	0.0017	0.0026	0.0019	0.0021	0.0017	0.0037	0.0016	0.0019	0.0021
Mean (F1-score)	**0.9276**	0.9259	0.9210	0.9231	0.9257	0.9242	0.9225	0.9253	0.9257	0.9237
SD (F1-score)	**0.0006**	0.0007	0.0016	0.0011	0.0009	0.0009	0.0021	0.0008	0.0008	0.0010
Mean (Specificity)	**0.9843**	0.9829	0.9798	0.9810	0.9826	0.9819	0.9812	0.9827	0.9833	0.9821
SD (Specificity)	**0.0015**	0.0016	0.0018	0.0017	0.0017	0.0015	0.0020	0.0015	0.0019	0.0019
Mean (Sensitivity)	**0.8771**	0.8751	0.8689	0.8716	0.8750	0.8729	0.8704	0.8743	0.8744	0.8720
SD (Sensitivity)	**0.0015**	0.0017	0.0026	0.0019	0.0021	0.0017	0.0037	0.0016	0.0019	0.0021
Mean (ROC_AUC)	**0.9657**	0.9651	0.9647	0.9652	0.9648	**0.9657**	0.9645	0.9654	0.9649	0.9648
SD (ROC_AUC)	**0.0009**	**0.0009**	0.0010	0.0010	**0.0009**	**0.0009**	**0.0009**	**0.0009**	0.0013	**0.0009**
Mean (MCC)	**0.8611**	0.8579	0.8488	0.8526	0.8576	0.8572	0.8523	0.8578	0.8576	0.8537
SD (MCC)	**0.0012**	0.0013	0.0029	0.0021	0.0015	0.0014	0.0027	0.0013	0.0019	**0.0012**

IBAVO-AO algorithm and well-known metaheuristic optimization techniques are compared and evaluated in terms of average accuracy, fitness values, number of selected features, Kappa, Precision, Recall, F1-score, Specificity, Sensitivity, ROC_AUC, and MCC. Note that boldface values denotes the best results. Regarding classification accuracy values presented in Table 6, the proposed IBAVO-AO algorithm succeeded in categorizing 92.7% of the news articles. After the proposed IBAVO-AO algorithm, BAVO ranked second with an average rating of 92.62% of news articles. Also, the stability of the proposed IBAVO-AO algorithm is relatively strong depending on the SD values of the different algorithms. Based on the number of features chosen, it is noted that the proposed IBAVO-AO algorithm comes first by selecting the minimum mean size of attributes on the used ISOT-FNs dataset, followed by the BSSA method. Additionally, the proposed IBAVO-AO algorithm obtained the best exploration capability over other algorithms regarding the mean selected features number, which was confirmed by choosing the least features number on the selected ISOT-FNs dataset. That verifies the capability of the proposed IBAVO-AO algorithm to neglect non-significant search regions and discover the most feasible regions. Therefore, the proposed IBAVO-AO algorithm can minimize the feature search region by identifying the most relevant attributes while preserving the highest classification accuracy. Based on fitness value. It should be noted that the proposed IBAVO-AO algorithm first obtains the minimum mean fitness value on the used ISOT-FNs dataset, followed by the BAVO and BAO methods. Finally, the proposed IBAVO-AO algorithm is based on the remaining evaluation measures.

Figure 4 reveals that the proposed approach on the selected attributes generated the highest performance compared to other optimization algorithms. The mean accuracy and F-measure of the proposed IBAVO-AO algorithm are 92.75% and 92.76%, respectively. Following the proposed IBAVO-AO algorithm, BAVO outperforms other known methods with average accuracy and F-measure of 92.62% and 92.62%. The BAO method also achieved nearly comparable outcomes with BAVO, producing accuracy and F-measure of 92.61% and 92.62%. The BSSA algorithm obtained good results by successfully classifying 92.58% of news articles.

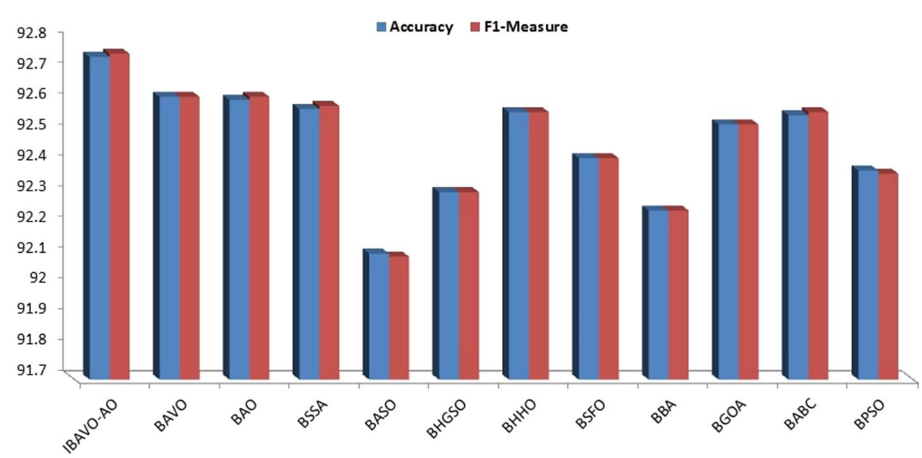


Fig. 4 Comparison of the proposed IBAVO-AO algorithm with other optimization algorithms for FND

The BHGSO method obtained the least results and classified 92.11% of news articles. The comparative study results show that the proposed IBAVO-AO algorithm obtained greater F-measure and accuracy than state-of-the-art optimization methods and was reliable in classifying various news articles.

Analysis of convergence

This section reveals an asymptotic investigation of the proposed IBAVO-AO algorithm for handling the FND strategy on the selected dataset to verify its capability in convergence, as shown in Fig. 5. These convergence graphs show the convergence capability of the proposed IBAVO-AO algorithm against their peers, which are all evaluated and executed under identical situations of the number of iterations and population size. Figure 5 shows that the proposed IBAVO-AO algorithm demonstrated fast yet optimal convergence behavior on the selected dataset. Hence, the proposed IBAVO-AO algorithm emphasizes its ability to acquire the optimal solution on time, ensuring an effective balance between exploration and exploitation capabilities.

Comparison results of the proposed IBAVO-AO algorithm against different algorithms from existing studies for FND

Table 7 illustrates the experimental outcomes of comparisons in terms of Accuracy, Kappa, Precision, Recall, and F1-score metrics between the suggested IBAVO-AO algorithm and other algorithms from existing studies, including the WOA-Xgb-Tree [24], AB [12], Wang-CNN [78], Wang-Bi-LSTM [78], Ridor [79, 80], and IBk [80, 81] for FND

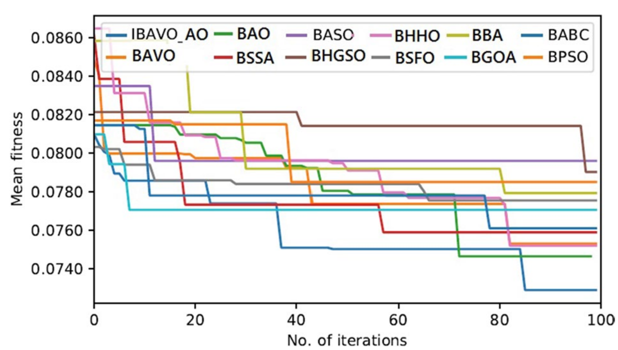


Fig. 5 Convergence graph of the proposed IBAVO-AO algorithm and other optimization algorithms on the selected dataset for FND

Table 7 Comparison of the proposed IBAVO-AO algorithm and other algorithms from existing studies for FND (Results are reported directly from the studies.)

Dataset	IBAVO-AO	WOA-Xgb-Tree [24]	AB [12]	Wang-CNN [78]	Wang-Bi-LSTM [78]	Ridor [79, 80]	IBk [80, 81]
Accuracy	**0.9275**	0.9186	0.9200	0.8700	0.8600	0.5570	0.5510
Kappa	**0.8555**	0.8370	N/A	N/A	N/A	N/A	N/A
Precision	**0.9843**	0.9230	0.9100	0.8400	0.9200	0.5630	0.5510
Recall	0.8771	**0.9200**	0.9100	0.9000	0.7800	0.5880	0.5510
F1-score	**0.9276**	0.9210	0.9100	0.8700	0.8400	0.5490	0.5500

issue. Note that boldface values denotes the best results. It can be noted that the proposed IBAVO-AO exceeded others in all performance measures except recall. Moreover, the WOA-Xgb-Tree algorithm is ranked first in recall and second in precision and F1-score. Finally, the IBK algorithm is ranked last in all performance measures.

Results analysis of the proposed IBAVO-AO algorithm versus various state-of-the-art ML methods and metaheuristic algorithms on new unseen datasets for FND

In this section, the proposed methodology is validated using different common datasets for detecting FNs. These datasets includes: FA-KES [82], BuzzFeed [83], UTK (Kaggle) [84], and Data (Kaggle) [85]. Two types of experiments are conducted on these datasets. First, a comparison with the most common ML models. Second, a comparison with recent optimization techniques. Table 8 shows the results of the proposed IBAVO-AO algorithm and other state-of-the-art ML methods on the new unseen datasets for FND regarding average accuracy, Kappa, Precision, Recall, and F1 score, where boldface values determine the best results. According to the results obtained in Table 8, the proposed IBAVO-AO algorithm outperformed all utilized ML models in all performance

Table 8 Results of the proposed IBAVO-AO algorithm and well-known ML methods using other datasets for FND

Dataset	Metric	IBAVO-AO	*k*-NN	SVM	DT	RF	GNB	MLP
FA-KES [82]	Accuracy	**0.6406**	0.4371	0.5364	0.4967	0.5166	0.4901	0.4172
BuzzFeed [83]	Accuracy	**0.8982**	0.3784	0.4595	0.5676	0.4865	0.5405	0.5135
UTK (Kaggle) [84]	Accuracy	**0.8331**	0.6196	0.5495	0.5896	0.6174	0.5150	0.6002
Data (Kaggle) [85]	Accuracy	**0.9445**	0.6965	0.6979	0.6925	0.7206	0.7019	0.7580
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	Kappa	**0.2809**	0.1258	0.0694	0.0093	0.0335	0.0214	0.1634
BuzzFeed	Kappa	**0.7913**	0.1869	0.0289	0.2128	0.0057	0.1465	0.0513
UTK (Kaggle)	Kappa	**0.6663**	0.2392	0.1017	0.1815	0.2364	0.0321	0.2001
Data (Kaggle)	Kappa	**0.8890**	0.3930	0.3957	0.3850	0.4412	0.4037	0.5160
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	Precision	**0.6471**	0.4342	0.5758	0.4889	0.5125	0.4828	0.4330
BuzzFeed	Precision	**0.8969**	0.4167	0.6667	0.7576	0.5714	0.7500	0.6000
UTK (Kaggle)	Precision	**0.8389**	0.6224	0.7361	0.8414	0.7679	0.5400	0.5915
Data (Kaggle)	Precision	**0.9239**	0.7712	0.7913	0.6607	0.6767	0.7475	0.7718
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	Recall	**0.6102**	0.4400	0.2533	0.2933	0.5467	0.3733	0.5600
BuzzFeed	Recall	**0.9317**	0.2381	0.0952	0.2381	0.3810	0.2857	0.4286
UTK (Kaggle)	Recall	**0.8264**	0.6155	0.1598	0.2247	0.3407	0.2271	0.6578
Data (Kaggle)	Recall	**0.9689**	0.5588	0.5374	0.7914	0.8449	0.6096	0.7326
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	F1-score	**0.6271**	0.4371	0.3519	0.3667	0.5290	0.4211	0.4884
BuzzFeed	F1-score	**0.9118**	0.3030	0.1667	0.3846	0.4571	0.4138	0.5000
UTK (Kaggle)	F1-score	**0.8325**	0.6189	0.2626	0.3547	0.4720	0.3198	0.6229
Data (Kaggle)	F1-score	**0.9458**	0.6481	0.6401	0.7202	0.7515	0.6716	0.7517
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4

Table 9 Results of the proposed IBAVO-AO algorithm and other optimization algorithms using other datasets for FND

Dataset	Metric	IBAVO-AO	BAVO	BAO	BSSA	BASO	BHGSO	BHHO	BSFO	BBA	BGOA	BABC	BPSO
FA-KES [82]	Accuracy	**0.6406**	0.6141	0.6163	0.6124	0.5627	0.5673	0.6088	0.5823	0.5746	0.6066	0.6128	0.5909
BuzzFeed [83]	Accuracy	**0.8982**	0.8514	0.8523	0.8541	0.7586	0.7739	0.8333	0.7919	0.7550	0.8279	0.8495	0.7964
UTK (Kaggle) [84]	Accuracy	**0.8331**	0.8303	0.8300	0.8286	0.8201	0.8265	0.8281	0.8300	0.8229	0.8279	0.8320	0.8259
Data (Kaggle) [85]	Accuracy	**0.9445**	0.9401	0.9394	0.9397	0.9322	0.9336	0.9381	0.9348	0.9309	0.9376	0.9399	0.9345
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	Fitness	**0.3598**	0.3865	0.3841	0.3881	0.4381	0.4338	0.3918	0.4183	0.4259	0.3941	0.3883	0.4098
BuzzFeed	Fitness	**0.1048**	0.1516	0.1507	0.1488	0.2438	0.2294	0.1697	0.2112	0.2473	0.1752	0.1540	0.2065
UTK (Kaggle)	Fitness	**0.1715**	0.1744	0.1747	0.1762	0.1845	0.1784	0.1765	0.1753	0.1814	0.1763	0.1739	0.1785
Data (Kaggle)	Fitness	**0.0598**	0.0642	0.0648	0.0649	0.0725	0.0718	0.0665	0.0703	0.0731	0.0670	0.0650	0.0698
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	Features Size	**20.2000**	22.4667	21.5000	21.9333	25.6333	27.0667	22.7667	23.9667	23.9333	23.1667	25.1000	24.2000
BuzzFeed	Features Size	**20.2333**	22.3000	22.3667	21.5667	23.6333	27.8333	23.5333	25.6667	23.6333	24.3333	25.3000	24.4333
UTK (Kaggle)	Features Size	31.3667	32.0000	31.9333	32.2667	31.9000	33.2667	31.4333	32.3200	30.2000	**29.8000**	38.2000	30.8000
Data (Kaggle)	Features Size	24.2667	24.3000	24.1667	26.1667	27.1667	30.2000	26.3000	28.4333	**23.3667**	26.0000	27.8333	24.6333
Ranking	W|T|L	**2|0|2**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	1|0|3	1|0|3	0|0|4	0|0|4
FA-KES	Kappa	**0.2809**	0.2279	0.2324	0.2243	0.1250	0.1342	0.2173	0.1644	0.1489	0.2129	0.2252	0.1816
BuzzFeed	Kappa	**0.7913**	0.6953	0.6972	0.7023	0.5096	0.5393	0.6583	0.5774	0.5014	0.6478	0.6917	0.5842
UTK (Kaggle)	Kappa	**0.6663**	0.6605	0.6599	0.6572	0.6402	0.6530	0.6561	0.6599	0.6458	0.6559	0.6641	0.6519
Data (Kaggle)	Kappa	**0.8890**	0.8801	0.8789	0.8794	0.8644	0.8671	0.8762	0.8695	0.8618	0.8751	0.8799	0.8690
Ranking	W|T|L	**4|0|0**	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4	0|0|4
FA-KES	Precision	**0.6471**	0.6204	0.6202	0.6192	0.5131	0.5694	0.6129	0.5834	0.5767	0.6112	0.6206	0.5935
BuzzFeed	Precision	**0.8969**	0.8587	0.8605	0.8730	0.7044	0.8042	0.8445	0.8262	0.7905	0.8412	0.8588	0.8196
UTK (Kaggle)	Precision	**0.8389**	0.8352	0.8340	0.8346	0.5602	0.8304	0.8331	0.8328	0.8281	0.8331	0.8349	0.8311
Data (Kaggle)	Precision	0.9239	0.9201	0.9201	0.9204	**0.9565**	0.9157	0.9190	0.9153	0.9119	0.9188	0.9217	0.9164

Table 9 (continued)

<table>
<tr><th>Dataset</th><th>Metric</th><th>IBAVO-AO</th><th>BAVO</th><th>BAO</th><th>BSSA</th><th>BASO</th><th>BHGSO</th><th>BHHO</th><th>BSFO</th><th>BBA</th><th>BGOA</th><th>BABC</th><th>BPSO</th></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>3|0|1</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>1|0|3</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
<tr><td>FA-KES</td><td>Recall</td><td>0.6102</td><td>0.5764</td><td>0.5880</td><td>0.5720</td><td>0.5324</td><td>0.5316</td><td>0.5782</td><td>0.5591</td><td>0.5427</td><td>0.5733</td><td>0.5720</td><td>0.5622</td></tr>
<tr><td>BuzzFeed</td><td>Recall</td><td>0.9317</td><td>0.8889</td><td>0.8889</td><td>0.8762</td><td>0.7762</td><td>0.8000</td><td>0.8714</td><td>0.8063</td><td>0.7794</td><td>0.8635</td><td>0.8857</td><td>0.8270</td></tr>
<tr><td>UTK (Kaggle)</td><td>Recall</td><td>0.8264</td><td>0.8246</td><td>0.8255</td><td>0.8212</td><td>0.8119</td><td>0.8223</td><td>0.8222</td><td>0.8138</td><td>0.8167</td><td>0.8218</td><td>0.8244</td><td>0.8199</td></tr>
<tr><td>Data (Kaggle)</td><td>Recall</td><td>0.9689</td><td>0.9639</td><td>0.9625</td><td>0.9627</td><td>0.9535</td><td>0.9551</td><td>0.9609</td><td>0.9582</td><td>0.9540</td><td>0.9600</td><td>0.9616</td><td>0.9563</td></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>4|0|0</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
<tr><td>FA-KES</td><td>F1-score</td><td>0.6271</td><td>0.5970</td><td>0.6031</td><td>0.5937</td><td>0.5466</td><td>0.5491</td><td>0.5943</td><td>0.5704</td><td>0.5583</td><td>0.5909</td><td>0.5940</td><td>0.5767</td></tr>
<tr><td>BuzzFeed</td><td>F1-score</td><td>0.9118</td><td>0.8711</td><td>0.8718</td><td>0.8714</td><td>0.7844</td><td>0.8002</td><td>0.8556</td><td>0.8141</td><td>0.7826</td><td>0.8502</td><td>0.8694</td><td>0.8215</td></tr>
<tr><td>UTK (Kaggle)</td><td>F1-score</td><td>0.8325</td><td>0.8298</td><td>0.8297</td><td>0.8278</td><td>0.8192</td><td>0.8263</td><td>0.8276</td><td>0.8277</td><td>0.8224</td><td>0.8274</td><td>0.8321</td><td>0.8254</td></tr>
<tr><td>Data (Kaggle)</td><td>F1-score</td><td>0.9458</td><td>0.9415</td><td>0.9408</td><td>0.9411</td><td>0.9336</td><td>0.9350</td><td>0.9395</td><td>0.9362</td><td>0.9324</td><td>0.9389</td><td>0.9412</td><td>0.9359</td></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>4|0|0</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
<tr><td>FA-KES</td><td>Specificity</td><td>0.6706</td><td>0.6513</td><td>0.6443</td><td>0.6522</td><td>0.5925</td><td>0.6026</td><td>0.6390</td><td>0.6053</td><td>0.6061</td><td>0.6395</td><td>0.6531</td><td>0.6193</td></tr>
<tr><td>BuzzFeed</td><td>Specificity</td><td>0.8542</td><td>0.8021</td><td>0.8042</td><td>0.8250</td><td>0.7354</td><td>0.7396</td><td>0.7833</td><td>0.7729</td><td>0.7229</td><td>0.7812</td><td>0.8021</td><td>0.7562</td></tr>
<tr><td>UTK (Kaggle)</td><td>Specificity</td><td>0.8400</td><td>0.8360</td><td>0.8344</td><td>0.8360</td><td>0.8284</td><td>0.8307</td><td>0.8339</td><td>0.8361</td><td>0.8291</td><td>0.8340</td><td>0.8347</td><td>0.8320</td></tr>
<tr><td>Data (Kaggle)</td><td>Specificity</td><td>0.9201</td><td>0.9162</td><td>0.9164</td><td>0.9167</td><td>0.9110</td><td>0.9120</td><td>0.9153</td><td>0.9113</td><td>0.9078</td><td>0.9152</td><td>0.9183</td><td>0.9127</td></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>4|0|0</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
<tr><td>FA-KES</td><td>Sensitivity</td><td>0.6102</td><td>0.5764</td><td>0.5880</td><td>0.5720</td><td>0.5324</td><td>0.5316</td><td>0.5782</td><td>0.5591</td><td>0.5427</td><td>0.5733</td><td>0.5720</td><td>0.5622</td></tr>
<tr><td>BuzzFeed</td><td>Sensitivity</td><td>0.9317</td><td>0.8889</td><td>0.8889</td><td>0.8762</td><td>0.7762</td><td>0.8000</td><td>0.8714</td><td>0.8063</td><td>0.7794</td><td>0.8635</td><td>0.8857</td><td>0.8270</td></tr>
<tr><td>UTK (Kaggle)</td><td>Sensitivity</td><td>0.8264</td><td>0.8246</td><td>0.8255</td><td>0.8212</td><td>0.8119</td><td>0.8223</td><td>0.8222</td><td>0.8138</td><td>0.8167</td><td>0.8218</td><td>0.8244</td><td>0.8199</td></tr>
<tr><td>Data (Kaggle)</td><td>Sensitivity</td><td>0.9689</td><td>0.9639</td><td>0.9625</td><td>0.9627</td><td>0.9535</td><td>0.9551</td><td>0.9609</td><td>0.9582</td><td>0.9540</td><td>0.9600</td><td>0.9616</td><td>0.9563</td></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>4|0|0</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
<tr><td>FA-KES</td><td>ROC_AUC</td><td>0.6177</td><td>0.5881</td><td>0.5919</td><td>0.5869</td><td>0.5389</td><td>0.5444</td><td>0.5778</td><td>0.5590</td><td>0.5431</td><td>0.5794</td><td>0.5809</td><td>0.5582</td></tr>
<tr><td>BuzzFeed</td><td>ROC_AUC</td><td>0.8942</td><td>0.8478</td><td>0.8600</td><td>0.8536</td><td>0.7735</td><td>0.7957</td><td>0.8364</td><td>0.8009</td><td>0.7641</td><td>0.8451</td><td>0.8560</td><td>0.7947</td></tr>
<tr><td>UTK (Kaggle)</td><td>ROC_AUC</td><td>0.9180</td><td>0.9162</td><td>0.9164</td><td>0.9159</td><td>0.9105</td><td>0.9146</td><td>0.9151</td><td>0.9145</td><td>0.9141</td><td>0.9143</td><td>0.9175</td><td>0.9138</td></tr>
</table>

Table 9 (continued)

<table>
<tr><th>Dataset</th><th>Metric</th><th>IBAVO-AO</th><th>BAVO</th><th>BAO</th><th>BSSA</th><th>BASO</th><th>BHGSO</th><th>BHHO</th><th>BSFO</th><th>BBA</th><th>BGOA</th><th>BABC</th><th>BPSO</th></tr>
<tr><td>Data (Kaggle)</td><td>ROC_AUC</td><td>0.9702</td><td>0.9683</td><td>0.9678</td><td>0.9680</td><td>0.9652</td><td>0.9677</td><td>0.9666</td><td>0.9663</td><td>0.9637</td><td>0.9679</td><td>0.9681</td><td>0.9665</td></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>4|0|0</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
<tr><td>FA-KES</td><td>MCC</td><td>0.2822</td><td>0.2288</td><td>0.2331</td><td>0.2256</td><td>0.1255</td><td>0.1348</td><td>0.2182</td><td>0.1649</td><td>0.1495</td><td>0.2138</td><td>0.2267</td><td>0.1823</td></tr>
<tr><td>BuzzFeed</td><td>MCC</td><td>0.7980</td><td>0.7020</td><td>0.7036</td><td>0.7095</td><td>0.5134</td><td>0.5426</td><td>0.6641</td><td>0.5815</td><td>0.5050</td><td>0.6527</td><td>0.6983</td><td>0.5879</td></tr>
<tr><td>UTK (Kaggle)</td><td>MCC</td><td>0.6664</td><td>0.6606</td><td>0.6600</td><td>0.6573</td><td>0.6404</td><td>0.6530</td><td>0.6562</td><td>0.6603</td><td>0.6459</td><td>0.6560</td><td>0.6641</td><td>0.6520</td></tr>
<tr><td>Data (Kaggle)</td><td>MCC</td><td>0.8901</td><td>0.8812</td><td>0.8799</td><td>0.8804</td><td>0.8653</td><td>0.8680</td><td>0.8772</td><td>0.8705</td><td>0.8627</td><td>0.8761</td><td>0.8807</td><td>0.8699</td></tr>
<tr><td>Ranking</td><td>W|T|L</td><td>4|0|0</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td><td>0|0|4</td></tr>
</table>

measures. None of the ML models used in the comparison ranked first in any of the performance measures.

In the second part of the analysis of the results, we compared the experimental results of the proposed IBAVO-AO algorithm with some known metaheuristic optimization algorithms on the new unseen datasets for FND, in which boldface values determine the best results. As shown in Table 9, the proposed IBAVO-AO algorithm and well-known metaheuristic optimization regarding average accuracy, fitness values, number of selected features, Kappa, Precision, Recall, F1-score, Specificity, Sensitivity, ROC_AUC, and MCC. According to the results obtained in Table 9, the proposed IBAVO-AO algorithm outperformed all utilized optimization techniques in all performance measures. None of the optimization techniques used in the comparison ranked first in any of the performance measures.

Conclusion and future work

Recently, FNs have been the most critical issue that harms society and individual users, making FND a great challenge. This study presented a new FNs classification and detection paradigm depending on an effective IBAVO-AO algorithm with the Xgb-Tree classifier. The proposed IBAVO-AO algorithm has preliminary stages: The ISOT-FNs dataset is retrieved first. Then, a pre-processing step is performed to transfer the unstructured data into structured data and analyze and extract the necessary attributes. This step includes extracting attributes from the ISOT-FNs dataset by ignoring useless words, stemming, tokenizing, encoding, and padding data into a sequence of integers using the GLOVE method; the extracted attributes are then filtered utilizing the effective Relief method to discover only suitable ones. Finally, the retrieved features are used to categorize the news items using the proposed IBAVO-AO based on The Xgb-Tree classifier. The suggested system obtained results have been analyzed and compared with state-of-the-art ML classifiers and optimization techniques concerning the accuracy, fitness values, the number of selected features, Kappa, Precision, Recall, F1-score, Specificity, Sensitivity, ROC_AUC, and MCC toward the same ISOT-FNs dataset. Moreover, one focus point was extracting attributes from news articles to assist the FND system in getting higher accuracy and shorter processing time. The results obtained from the proposed IBAVO-AO algorithm showed that the extracted attributes positively affect the performance of the proposed FND system.

Following this, it's important to state the limitations of the study to provide a balanced and realistic view of its scope and applicability:

- Dataset Scope and Diversity: The ISOT-FNs dataset, while comprehensive, may not fully encompass the broad spectrum of FNs sources, especially more subtle or complex misinformation strategies.
- Single-Modality Focus: The study focused solely on text-based news articles, excluding multimedia elements like images or videos often integral to FNs.
- Algorithmic Adaptability: The performance and adaptability of the IBAVO-AO algorithm across various datasets and types of FNs content require further exploration.

In the future, we tend to analyze and investigate these topics:

- Incorporating Multi-modal Data: Future research will focus on processing news articles that include images and text, moving beyond the text-only approach to provide a more comprehensive analysis of FNs.
- Exploring Diverse Classification Methods: Plans include applying other classification methods like neural networks, k-NN, and Random Forest (RF) to assess further the behavior and efficacy of the IBAVO-AO algorithm in various classification tasks.
- Broadening Input Features: We aim to analyze optimization methods with multiple input features, ranging from raw text to handcrafted attributes. This approach could uncover new insights and enhance the system's ability to detect FNs more accurately.

Acknowledgements
Not applicable.

Author contributions
Conceptualization, methodology, software, statistical analysis, data analysis, literature review, discussion, writing—original draft preparation: ABE, AAA, and KMH; data downloading: AHA, ABE, and AAA; writing—review and editing: ABE, AAA, and KMH; visualization: AHA, AAA, and ABE; supervision: KMH. All authors read and approved the final manuscript.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and materials
The developed software and code in this study are available on request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 24 June 2023 Accepted: 12 February 2024
Published online: 19 March 2024

References

1. Bessi A, Coletto M, Davidescu GA, Scala A, Caldarelli G, Quattrociocchi W. Science vs conspiracy: collective narratives in the age of misinformation. PLoS ONE. 2015;10(2):e0118093.
2. Gravanis G, Vakali A, Diamantaras K, Karadais P. Behind the cues: a benchmarking study for fake news detection. Expert Syst Appl. 2019;128:201–13.
3. Alzanin SM, Azmi AM. Detecting rumors in social media: a survey. Procedia Comput Sci. 2018;142:294–300.
4. Shu K, Sliva A, Wang S, Tang J, Liu H. Fake news detection on social media: a data mining perspective. ACM SIGKDD Explorations Newsl. 2017;19(1):22–36.
5. Bondielli A, Marcelloni F. A survey on fake news and rumour detection techniques. Inf Sci. 2019;497:38–55.
6. Ferrara E, Varol O, Davis C, Menczer F, Flammini A. The rise of social bots. Commun ACM. 2016;59(7):96–104.
7. Zhang X, Ghorbani AA. An overview of online fake news: characterization, detection, and discussion. Inf Proc Manag. 2020;57(2):102025.
8. Della Vedova ML, Tacchini E, Moret S, Ballarin G, DiPierro M, de Alfaro L, Automatic online fake news detection combining content and social signals, In: 2018 22nd conference of open innovations association (FRUCT), IEEE, 2018;272–9.
9. Ghosh P, Azam S, Jonkman M, Karim A, Shamrat FJM, Ignatious E, Shultana S, Beeravolu AR, De Boer F. Efficient prediction of cardiovascular disease using machine learning algorithms with relief and lasso feature selection techniques. IEEE Access. 2021;9:19304–26.
10. Shamrat FJM, Tasnim Z, Ghosh P, Majumder A, Hasan MZ, Personalization of job circular announcement to applicants using decision tree classification algorithm, in: 2020 IEEE International Conference for Innovation in Technology (INOCON), IEEE, 2020;1–5.

11. Afrin S, Shamrat FJM, Nibir TI, Muntasim MF, Moharram MS, Imran M, Abdulla M. Supervised machine learning based liver disease prediction approach with lasso feature selection. Bull Electric Eng Inf. 2021;10(6):3369–76.
12. Nasir JA, Khan OS, Varlamis I. Fake news detection: a hybrid cnn-rnn based deep learning approach. Int J Inf Manag Data Insights. 2021;1(1):100007.
13. Sahoo SR, Gupta BB. Multiple features based approach for automatic fake news detection on social networks using deep learning. Appl Soft Comput. 2021;100:106983.
14. Choudhary A, Arora A. Linguistic feature based learning model for fake news detection and classification. Expert Syst Appl. 2021;169:114171.
15. Singhania S, Fernandez N, Rao S. 3han: A deep neural network for fake news detection, in: International conference on neural information processing, Springer, 2017;572–581.
16. Kaliyar RK, Goswami A, Narang P, Sinha S. Fndnet-a deep convolutional neural network for fake news detection. Cogn Syst Res. 2020;61:32–44.
17. Umer M, Imtiaz Z, Ullah S, Mehmood A, Choi GS, On B-W. Fake news stance detection using deep learning architecture (cnn-lstm). IEEE Access. 2020;8:156695–706.
18. Rodríguez ÁI, Iglesias LL. Fake news detection using deep learning, arXiv preprint arXiv:1910.03496.
19. Monti F, Frasca F, Eynard D, Mannion D, Bronstein MM. Fake news detection on social media using geometric deep learning, arXiv preprint arXiv:1902.06673.
20. Thota A, Tilak P, Ahluwalia S, Lohia N. Fake news detection: a deep learning approach. SMU Data Sci Rev. 2018;1(3):10.
21. Abedalla A, Al-Sadi A, Abdullah M. A closer look at fake news detection: a deep learning perspective, in: Proceedings of the 2019 3rd International Conference on Advances in Artificial Intelligence, 2019;24–28.
22. Ozbay FA, Alatas B. A novel approach for detection of fake news on social media using metaheuristic optimization algorithms. Elektronika ir Elektrotechnika. 2019;25(4):62–7.
23. Ozbay FA, Alatas B. Adaptive salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media. Multimedia Tools Appl. 2021;80(26):34333–57.
24. Sheikhi S. An effective fake news detection method using woa-xgbtree algorithm and content-based features. Appl Soft Comput. 2021;109:107559.
25. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014;1532–1543.
26. Jain V, Kaliyar RK, Goswami A, Narang P, Sharma Y. Aenet: an attention-enabled neural architecture for fake news detection using contextual features, Neural Computing and Applications 2021;1–12.
27. Abd El-Mageed AA, Gad AG, Sallam KM, Munasinghe K, Abohany AA. Improved binary adaptive wind driven optimization algorithm-based dimensionality reduction for supervised classification. Comput Ind Eng. 2022;167:107904.
28. Abd El-Mageed AA, Abohany AA, Saad HM, Sallam KM. Parameter extraction of solar photovoltaic models using queuing search optimization and differential evolution. Appl Soft Comput 2023;110032.
29. Vishwakarma DK, Varshney D, Yadav A. Detection and veracity analysis of fake news via scrapping and authenticating the web search. Cogn Syst Res. 2019;58:217–29.
30. Castillo C, Mendoza M, Poblete B. Information credibility on twitter, in: Proceedings of the 20th international conference on World wide web, 2011;675–684.
31. Jin F, Dougherty E, Saraf P, Cao Y, Ramakrishnan N. Epidemiological modeling of news and rumors on twitter, in: Proceedings of the 7th workshop on social network mining and analysis, 2013;1–9.
32. Wu K, Yang S, Zhu KQ, False rumors detection on sina weibo by propagation structures, in: 2015 IEEE 31st international conference on data engineering, IEEE, 2015;651–62.
33. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M. Detecting rumors from microblogs with recurrent neural networks., in: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), 2016;3818–3824.
34. Sampson J, Morstatter F, Wu L, Liu H. Leveraging the implicit structure within social media for emergent rumor detection, in: Proceedings of the 25th ACM international on conference on information and knowledge management, 2016;2377–2382.
35. Yang F, Liu Y, Yu X, Yang M. Automatic detection of rumor on sina weibo, in: Proceedings of the ACM SIGKDD workshop on mining data semantics, 2012;1–7.
36. Reganti AN, Maheshwari T, Kumar U, Das A, Bajpai R. Modeling satire in English text for automatic detection, in: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), IEEE, 2016;970–977.
37. Buschmeier K, Cimiano P, Klinger R. An impact analysis of features in a classification approach to irony detection in product reviews, in: Proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis, 2014;42–49.
38. Kwon S, Cha M, Jung K. Rumor detection over varying time windows. PLoS ONE. 2017;12(1):e0168344.
39. Sedik A, Abohany AA, Sallam KM, Munasinghe K, Medhat T. Deep fake news detection system based on concatenated and recurrent modalities. Expert Syst Appl. 2022;208:117953.
40. Meel P, Vishwakarma DK. A temporal ensembling based semi-supervised convnet for the detection of fake news articles. Expert Syst Appl. 2021;177:115002.
41. Kumar S, Asthana R, Upadhyay S, Upreti N, Akbar M. Fake news detection using deep learning models: a novel approach. Trans Emerg Telecommun Technol. 2020;31(2):e3767.
42. Shim J-S, Lee Y, Ahn H. A link2vec-based fake news detection model using web search results. Expert Syst Appl. 2021;184:115491.
43. Zervopoulos A, Alvanou AG, Bezas K, Papamichail A, Maragoudakis M, Kermanidis K, Deep learning for fake news detection on twitter regarding the 2019 Hong Kong protests. Neural Comput Appl. 2021;1–14.

44. Huang Y-F, Chen P-H. Fake news detection using an ensemble learning model based on self-adaptive harmony search algorithms. Expert Syst Appl. 2020;159:113584.
45. Sansonetti G, Gasparetti F, D'aniello G, Micarelli A. Unreliable users detection in social media: Deep learning techniques for automatic detection. IEEE Access. 2020;8:213154–67.
46. Samadi M, Mousavian M, Momtazi S. Deep contextualized text representation and learning for fake news detection. Inf Proc Manag. 2021;58(6):102723.
47. Khan JY, Khondaker MTI, Afroz S, Uddin G, Iqbal A. A benchmark study of machine learning models for online fake news detection. Mach Learning Appl. 2021;4:100032.
48. Kira K, Rendell LA et al. The feature selection problem: traditional methods and a new algorithm, in: Aaai, 1992;2:129–134.
49. Kononenko I. Estimating attributes: analysis and extensions of relief, in: European conference on machine learning, Springer, 1994;171–182.
50. Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng. 2021;158:107408.
51. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH. Aquila optimizer: a novel metaheuristic optimization algorithm. Comput Ind Eng. 2021;157:107250.
52. Steenhof K, Kochert MN, Mcdonald TL. Interactive effects of prey and weather on golden eagle reproduction. J Animal Ecol. 1997;66(3):350–62.
53. Chen T, Guestrin C. Xgboost: a scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016;785–794.
54. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Ann Stat. 2000;28(2):337–407.
55. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;1189–1232.
56. Friedman JH. Stochastic gradient boosting. Comput Stat Data Anal. 2002;38(4):367–78.
57. Isot fake news dataset, https://www.uvic.ca/engineering/ece/isot/datasets/fake-news.
58. Rokach L, Maimon O. Data mining with decision trees: theory and applications. World Scientific; 2020.
59. Zhang P, Zhou D. Understanding the k-nearest neighbor: from an algebraic perspective. Pattern Recogn. 2020;100:107149.
60. Webb GI, Keogh E, Miikkulainen R. Naive bayes: the good, the bad, and the ugly, in: Advances in Intelligent Data Analysis XVII, Springer, 2019;428–440.
61. Liu Q, Zhou Z-H. Support vector machines: theory, algorithms, and extensions. CRC Press; 2019.
62. Belgiu M, Drăguţ L. Random forest in remote sensing: a review of applications and future directions. ISPRS J Photogramm Remote Sens. 2016;114:24–31.
63. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.
64. Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng. 2020;8(1):22–34.
65. Zhao W, Wang L, Zhang Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl-Based Syst. 2019;163:283–304.
66. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: a novel physics-based algorithm. Futur Gener Comput Syst. 2019;101:646–67.
67. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst. 2019;97:849–72.
68. Shadravan S, Naji HR, Bardsiri VK. The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng Appl Artif Intell. 2019;80:20–34.
69. Mirjalili S, Mirjalili SM, Yang X-S. Binary bat algorithm. Neural Comput Appl. 2014;25(3):663–81.
70. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I. Grasshopper optimization algorithm for multi-objective optimization problems. Appl Intell. 2018;48(4):805–20.
71. Karaboga D, Basturk B. On the performance of artificial bee colony (abc) algorithm. Appl Soft Comput. 2008;8(1):687–97.
72. Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Swarm Intell. 2007;1(1):33–57.
73. Yin M, Wortman Vaughan J, Wallach H. Understanding the effect of accuracy on trust in machine learning models, in: Proceedings of the 2019 chi conference on human factors in computing systems, 2019;1–12.
74. De Medeiros AKA, Guzzo A, Greco G, Van Der Aalst WM, Weijters A, Van Dongen BF, Saccà D. Process mining based on clustering: A quest for precision, in: International Conference on Business Process Management, Springer, 2007;17–29.
75. Amigó E, Gonzalo J, Artiles J, Verdejo F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. Inf Retrieval. 2009;12(4):461–86.
76. Amigó E, Gonzalo J, Artiles J, Verdejo F. Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks. J Artif Intel Res. 2011;42:689–718.
77. Parikh R, Mathai A, Parikh S, Sekhar GC, Thomas R. Understanding and using sensitivity, specificity and predictive values. Indian J Ophthalmol. 2008;56(1):45.
78. Ahmad I, Yousaf M, Yousaf S, Ahmad MO. Fake news detection using machine learning ensemble methods. Complexity. 2020;2020:1–11.
79. Lakmali K, Haddela PS. Effectiveness of rule-based classifiers in sinhala text categorization, in: 2017 National Information Technology Conference (NITC). IEEE. 2017;153–8.

80. Ozbay FA, Alatas B. Fake news detection within online social media using supervised artificial intelligence algorithms. Physica A. 2020;540:123174.
81. Kaladhar D, Pottumuthu BK, Rao PVN, Vadlamudi V, Chaitanya AK, Reddy RH. The elements of statistical learning in colon cancer datasets: data mining, inference and prediction. Algorithms Res. 2013;2(1):8–17.
82. Fa-kes fake news dataset, https://zenodo.org/records/2607278.
83. Buzzfeed fake news dataset, https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook#.gt1ygzDN.
84. Utk fake news dataset, https://www.kaggle.com/c/fake-news.
85. Data fake news dataset, https://www.kaggle.com/datasets/jruvika/fake-news-detection.