

Article

Intelligent Detection of False Information in Arabic Tweets Utilizing Hybrid Harris Hawks Based Feature Selection and Machine Learning Models

Thaer Thaher ^{1,*} , Mahmoud Saheb ² , Hamza Turabieh ³  and Hamouda Chantar ⁴ 

¹ Department of Engineering and Technology Sciences, Arab American University, P.O. Box 240 Jenin, 13 Zababdeh, Palestine

² IT and Computer Engineering College, Palestine Polytechnic University, P.O. Box 198 Hebron, Palestine; alsaheb@ppu.edu

³ Department of Information Technology, Collage of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; h.turabieh@tu.edu.sa

⁴ Faculty of Information Technology, Sebha University, Sebha 18758, Libya; ham.chantar@sebha.edu.ly

* Correspondence: t.thaher@student.aaup.edu

Abstract: Fake or false information on social media platforms is a significant challenge that leads to deliberately misleading users due to the inclusion of rumors, propaganda, or deceptive information about a person, organization, or service. Twitter is one of the most widely used social media platforms, especially in the Arab region, where the number of users is steadily increasing, accompanied by an increase in the rate of fake news. This drew the attention of researchers to provide a safe online environment free of misleading information. This paper aims to propose a smart classification model for the early detection of fake news in Arabic tweets utilizing Natural Language Processing (NLP) techniques, Machine Learning (ML) models, and Harris Hawks Optimizer (HHO) as a wrapper-based feature selection approach. Arabic Twitter corpus composed of 1862 previously annotated tweets was utilized by this research to assess the efficiency of the proposed model. The Bag of Words (BoW) model is utilized using different term-weighting schemes for feature extraction. Eight well-known learning algorithms are investigated with varying combinations of features, including user-profile, content-based, and words-features. Reported results showed that the Logistic Regression (LR) with Term Frequency-Inverse Document Frequency (TF-IDF) model scores the best rank. Moreover, feature selection based on the binary HHO algorithm plays a vital role in reducing dimensionality, thereby enhancing the learning model's performance for fake news detection. Interestingly, the proposed BHHO-LR model can yield a better enhancement of 5% compared with previous works on the same dataset.

Keywords: false information; natural language processing; machine learning; feature selection; meta-heuristics; Twitter



Citation: Thaher, T.; Saheb, M.; Turabieh, H.; Chantar, H. Intelligent Detection of False Information in Arabic Tweets Utilizing Hybrid Harris Hawks Based Feature Selection and Machine Learning Models. *Symmetry* **2021**, *13*, 556. <https://doi.org/10.3390/sym13040556>

Academic Editor: Peng-Yeng Yin

Received: 16 February 2021

Accepted: 23 March 2021

Published: 27 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The world is currently witnessing the era of huge information where data generated by social media and social network sites (SNSs) are increasing rapidly. SNSs are electronic media where millions of people daily exchange a huge amount of information in digital text format. Twitter is recognized as one of the most widely used social media networks. Social media networks form a vast source of worthy information that attracts scholars in the field of knowledge discovery for developing various techniques and discovering potentially useful hidden information. Unfortunately, part of social media users misuse the network by spreading fake or false information for political and financial aims, which may hurt the public.

In the social media mining field, data mining and text mining techniques are exploited in conjunction with each of social network analysis and information retrieval for gaining

useful hidden data that uncover the purpose of spreading news, sentiments, or opinions over social media platforms [1,2]. Understanding or monitoring the sentiment tendency of people is essential for making the proper decision in organizations. Sentiment Analysis (SA) or opinion mining is the process of analyzing the sentiment, feeling, and judgment of people regarding many things such as service, product, organization, or person [3]. Essentially, SA is the task of analyzing a collection of natural language texts to extract features that ideally characterize an object like a person, service, or organization. Sentiment or opinion in SA is differentiated by key components comprising the orientation of sentiment, the object, the aspect, the time, and the opinion holder [4]. The object could be a person, event, organization, or service, whereas the aspect means the object's attribute. Sentiment orientation refers to the polarity of sentiment, which can be positive, negative, or neutral. The opinion holder is an organization or person that expresses an opinion towards an object at a particular moment [4]. SA can be examined at different levels, including aspect level, sentence level, and document level. Each piece of text in document level SA is considered as a separate unit that bears an opinion regarding a particular object. In the sentence level of SA, the sentence is investigated to determine whether it expresses an opinion or not. In contrast, the aspect level of SA indicates the identification of key components with the assertion on two tasks, including aspect extraction and aspect SA [3,4].

SA is typically performed using three methods, including supervised methods, unsupervised methods, or a combination of supervised and unsupervised methods. Based on labeled data, the supervised approach uses an ML classifier such as Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Decision Trees (DT) to build a classification model for predicting the sentiment of a given text. In contrast, the unsupervised approach relies on a lexicon to determine the sentiment of words [4]. Fake or false information is one of the main SA applications that aim to detect fake information in social media. Fake news in social media networks can be defined as any published and shared unverifiable news. Fake news may contain rumors, propaganda, or misleading information about a person, organization, or service [5]. Several efforts have been made to develop methods for detecting fake news in social media networks such as Twitter and Facebook. Broadly, fake news detection methods can be content-based, or user-based [5]. In content-based, fake news is detected from texts [6] or images [7] while user-based methods depend on bio information [8] and activity tracing [9].

The Arabic language has the fourth position among used languages on the world wide web. It is also rated as the sixth language used on Twitter [10]. The Arabic language is the formal spoken language of around two billion Muslims worldwide, and it is the language of Islam's holy book. Arabic has a rich morphology that makes its automatic processing a challenging task for NLP techniques [11]. Unlike English, the Arabic language is written from right to left. Furthermore, it has various writing styles, including the Modern Standard Arabic (MSA), which is formal Arabic, the Classical Arabic that represents the language of the holy book of Islam, and the Dialectal Arabic (DA) [10,12].

SA and its applications such as fake news detection can be viewed as a binary text classification task where a given text (e.g., a tweet) is labeled as fake or not. To achieve a text processing task, a text is ideally converted into a term-frequency vector. Such conversion results in high dimensionality of feature space because every distinct word in each text would appear as a dimension in the feature space. Consequently, the efficacy of classification models is remarkably degraded due to the high dimension of features and the presence of noisy features. Therefore, it is demandable to lessen the number of features to increase the classification accuracy [12]. Feature Selection (FS) is one of the dimensionality reduction approaches that aim to discard noisy and irrelevant features while preserving highly associated features with the classification process [12].

Generally speaking, FS follows two branches: filter and wrapper. Filter FS approaches often exploit a statistical measure to allocate weight to every feature in the feature space. Any feature that has a weight less than a pre-specified threshold is considered an uninformative feature and removed from the feature space. Patterns of filter approaches include

the Chi-squared test, information gain, and correlation coefficient [12–14]. The wrapper FS approach treats the selection of a subset of features like a search optimization problem that aims to find an ideal subset of features. In this context, a prediction model is used to evaluate several subsets of features produced by a search algorithm to discover the best subset of features. In general, wrapper FS, compared to filter FS, can achieve better in terms of accuracy since it can expose and utilize dependencies between selected candidates in a subset of features [12,13].

The literature reveals that several recently founded meta-heuristic optimization algorithms have been applied as wrapper FS approaches. Instances of these algorithms include Slap Swarm Algorithm (SSA) [15], Moth-Flame Optimization (MFO) [16], Dragonfly Algorithm (DA) [17], Ant Lion Optimization (ALO) [18], and Whale Optimization Algorithm (WOA) [19]. Recently, Heidari and his co-authors introduced a novel meta-heuristic optimizer called Harris Hawks Optimization (HHO) [20]. It mimics the hawks' behavior when they suddenly attack prey from several directions. Many recent research papers have confirmed the excellent performance of HHO in dealing with real optimization problems in various domains such as photovoltaic models, drug design and discovery, image segmentation, feature selection, and job scheduling [21]. HHO has several features that make it superior to other swarm-based algorithms; it is flexible, easy to use, and complete. It is free of internal parameters that need to be tuned. In addition, the varying exploration and exploitation styles with greedy schemes contribute to achieving high-quality results and favorable convergence [22]. The original form of HHO was introduced to solve real-valued optimization problems. In this paper, to be used as a wrapper FS approach for searching the optimal subset of features for Arabic fake news identification, HHO was combined with fuzzy V-shaped transfer functions to enable it to work in a binary search space.

This paper introduces a hybrid classification framework for the early detection of fake news in Arabic tweets. The proposed model is developed based on the integration of NLP techniques and the HHO-based wrapper FS scheme. The key contributions of this research are summarized as follows:

- Conducting different feature extraction models (i.e., TF, BTF, and TF-IDF) and eight well-known ML algorithms to identify suitable models for detecting fake news in Arabic tweets.
- Adapting a binary variant of the HHO algorithm, for the first time in the text classification domain, with the sake of eliminating the irrelevant/redundant features and enhancing the performance of classification models. HHO has been recently utilized as an FS approach to deal with complex datasets by the first author of this work and his co-authors in [22,23]. HHO has shown its superiority compared to several recent optimization algorithms such as Binary Gray Wolf Optimization (BGWO) and Binary WOA in the FS domain. For this reason, it has been nominated in this work to tackle the problem of FS in Arabic text classification.
- The efficiency of the proposed model has been verified by comparing it with other state-of-the-art approaches and has shown promising results.

This paper's structure is set as follows: Section 2 explores the related works of SA and fake news detection. Section 3 is devoted to NLP techniques and the HHO algorithm's theoretical background. Section 4 presents the proposed methodology in detail. Section 5 reports results and comparative evaluations. Finally, Section 6 concludes the overall work and presents the future trends.

2. Review of Related Works

2.1. Sentiment Analysis

As stated in [24], Elouardighi et al. introduced an ML-based SA approach for Arabic comments on Facebook, where n-grams, TF, and TF-IDF were applied as term weighting schemes for features extraction. It was observed that the results of SVM are better compared to Naive Bayes (NB) and Random Forest (RF) classifiers. Moreover, Biltawi et al. [25] introduced a hybrid approach for the Arabic sentiment classification problem where lexicon-

based and corpus-based techniques were combined depending on Twitter and a movie reviews dataset (OCR). The results showed that the proposed approach in conjunction with RF achieved better than the results obtained from a corpus-based approach. In addition, Daoud et al. [26] proposed an approach for clustering news documents where the seeds of clusters were selected using k-means with a Particle Swarm Optimization (PSO) algorithm. The results revealed that the proposed news clustering approach, along with applying light stemming, yielded better results in terms of F-measure. However, Tubishat et al. [3] proposed an approach based on an improved version of the WOA for the selection of the best subset of features for Arabic SA. The proposed approach enhanced the Arabic sentiment classification in terms of accuracy rates. Al-Ayyoub et al. [27] conducted a comprehensive survey to apply deep learning to tackle various Arabic natural processing problems such as Arabic text classification, optical character recognition, and detection of spam opinion, and SA. Al-Azani performed sentiment classification of Arabic tweets into positive or negative, and El-Alfy [28], in which textual features were combined with Emojis. Different techniques comprising Bow, TF-IDF, latent semantic analysis, and two variants of Word embedding techniques were applied for feature extraction. In terms of F-measure, the word embedding technique that combines Emojis and skip-gram was superior when evaluated using an SVM classifier.

In literature, SA has been investigated for other languages such as English. For example, Rout et al. [1] developed a sentiment and emotion analysis approach for classifying social media texts. An unsupervised method was proposed for identifying sentiment for tweets. In addition, the authors also applied supervised techniques, including multinomial NB and SVM, to determine sentiment. It was observed that the unsupervised approach was the best in terms of accuracy. The authors in [29] claimed that one of the main weaknesses of SA is caused by a deficiency of learning algorithms to generalize the domain of given text, which results in what is called the domain adaptation problem. Therefore, they introduced an evolutionary ensemble approach for SA and found that the ensemble of SA methods is an appropriate solution for dealing with domain adaptation problems. Vizcarra et al. [30] proposed a SA approach for Spanish tweets, where a model based on various word embedding approaches and Convolutional Neural Networks (CNN) was used to predict the polarity of the sentiment.

2.2. False Information Detection

In the Arabic context, there is an absence of research studies that tackle the problem of fake news detection on online social media platforms such as Twitter and Facebook. However, Jardaneh et al. [31] proposed a supervised ML-based model for identifying fake news in Arabic tweets where content-based, user-based, and SA based features were used for data representation. In addition, several ML classifiers, including RF, DT, AdaBoost, and LR, were used for building the classification model. Results reveal that the proposed system can identify fake news in Arabic tweets with a classification accuracy of 76%. Al-Khalifa et al. [32] developed a system to assess the credibility of news content published in tweets. The proposed system allocates one of three credibility levels, low, medium, and high, to each tweet. Two approaches were used to determine the level of credibility. One of these approaches depends on trusted sources of news to find links between these sources and a given tweet, while the second approach uses a set of features alongside the results of the first approach. However, results show that allocating credibility levels to tweets using the first approach is better in terms of precision and recall. Sabbeh et al. [33] proposed a ML-based model that exploits topic and user-related features to measure the credibility of Arabic news obtained from Twitter. In addition, several classifiers, including DT, SVM, and NB, were utilized. Results show that the proposed approach outperformed two approaches in the literature.

In contrast, several efforts have been made for fake news identification in the English language context. For example, Ajao et al. [6] proposed a hybrid approach based on CNN and Long short-term memory (LSTM) recurrent neural network models to identify

and classify fake news in posted tweets. The proposed approach achieved 82% accuracy. A conditional random field (CRF) classifier boosted with the context preceding a tweet was applied by [34] for identifying rumors from tweets. The introduced approach achieved better performance when compared against a method that depends on matching query posts with a set of regular expressions. Natali et al. [35] introduced a model for fake news detection that combines three modules. The first one that is based on Recurrent Neural Network (RNN) is used to determine the temporal pattern of user activity on a given text, while the second module depends on the behavior of users. The two modules were integrated with a third one and used to identify whether a given article is fabricated or not.

In summary, the inspected related works confirmed the efficiency of supervised ML approaches for identifying sentiment in tweets. However, there is a lack of benchmark datasets on fake news, especially in the Arabic language context. In the Arabic context, the state of fake news detection on online social media platforms is still rudimentary and open for conducting more research studies. Moreover, the high dimensionality of feature space in the text classification domain may remarkably degrade the classification model's efficacy due to the presence of irrelevant and noisy features. These facts motivated us to propose an efficient classification model for the early detection of fake news in Arabic tweets adapting the recent HHO algorithm (which, to the best of our knowledge, has not been investigated in this domain yet) as a feature selection approach.

3. Theoretical Background (Preliminaries)

3.1. Natural Language Processing (NLP)

NLP is recognized as a sub-field of Artificial Intelligence (AI) concerns processing and analyzing numerous amounts of natural language data such as speech and text. For instance, in the case of text, the main objective is to transform unstructured text into a structured form that can be analyzed for extracting useful knowledge. NLP comprises different techniques for processing and structuring sentiment data. Three techniques were mainly involved in this research: tokenization, linguistic modules (normalization, removing stop words, stemming), and text vectorization.

3.1.1. Tokenization and Linguistic Modules

Tokenization is an essential step in NLP processing tasks, as the text is chopped up into smaller pieces, such as individual terms or words, called *tokens* for further SA purposes [36]. For instance, tokenizing the sentence (S) "detection of false information is one of the main sentiment analysis applications" yields the result $S = ['\text{detection}', '\text{of}', '\text{false}', '\text{information}', '\text{is}', '\text{one}', '\text{of}', '\text{the}', '\text{main}', '\text{sentiment}', '\text{analysis}', '\text{applications}']$. However, some very common words in a language (called stop words) are irrelevant and can be filtered out without sacrificing the meaning [37]. Accordingly, after removing stop words, the sentence S becomes $['\text{detection}', '\text{false}', '\text{information}', '\text{main}', '\text{sentiment}', '\text{analysis}', '\text{applications}']$.

Furthermore, normalization is a fundamental preprocessing step. Text normalization is defined as transforming non-standard words to their predefined (or standard) forms [38]. Examples of normalization rules that can be applied in the Arabic language are: converting Alef character (أ, إ, ئ) to (a), normalize Hamza (آ, إ, ؤ) to (a), and strip Arabic diacritics. Another useful normalization technique is called stemming. It is a heuristic process that aims to reduce inflected and derived words into their stem that is affixed to prefixes and suffixes or to reduce them to word roots form, known as a lemma. There are various types of stemming algorithms customized according to the text language. Examples include Porter stemmer [39] for English and the Information Science Research Institute's (ISRI) Arabic stemmer [40].

3.1.2. Text Vectorization

Text vectorization, or more simply, feature extraction, is the process of converting the unstructured text into structured representation (numerical features) such that ML algorithms can be applied for mining and knowledge extraction purposes [2]. In simple

terms, the numerical features are extracted utilizing words-based statistical measurements. Therefore, the text is viewed as a multivariate sample (vector) of such measures. Bag of Words (BoW) is a common text representative model that is widely utilized for text classification applications [2,12,31]. It is a flexible and straightforward textual representation scheme that describes the occurrences or count of terms within a piece of text without considering the exact ordering and semantic structure. In this model, different methods are used in scoring the occurrences of words (i.e., term-weighting). The prominent techniques are Term Frequency (TF), Binary Term Frequency (BTF), and TF-IDF. To determine the appropriate text representation model among the above-mentioned ones, particularly for Arabic fake information detection, the three techniques are utilized in this research.

3.1.3. Term Frequency-Based Model

In this technique, each term t is assigned a weight that represents the number of occurrences (i.e., frequency) of that term in a particular text d [41]. Therefore, the text is viewed as a word-count vector. However, the relevance of the term does not grow proportionally with its frequency [36]. To handle this issue, normalization-based TF techniques have been utilized, such as the logarithm function as in Equation (1):

$$w_t = \begin{cases} 1 + \log_{10} TF_t & TF_t > 0 \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

where w_t is the weighted frequency of the term t in a given text, and TF_t denotes the natural term frequency of t .

Another variant of TF based textual representation is called BTF. In this technique, the text is represented as a binary vector that refers to the absence (0) or presence (1) of the term in the text.

The major disadvantage of traditional frequency-based weighting techniques is that all terms are handled with equal importance. Specific frequent terms are irrelevant (less informative) compared to rare terms. Therefore, we need an efficient measure that scores frequent terms with high weights but lower than for rare terms. For this purpose, other measures such as Document Frequency (DF), Inverse Document Frequency (IDF), and TF-IDF were introduced for term weighting.

3.1.4. Term Frequency-Inverse Document Frequency Scheme

Document frequency of a term (DF_t) is defined as the number of documents (texts) in the corpus containing the term, whereas the IDF of a term (IDF_t) is a weight that represents how common the term is used; it can be calculated using Equation (2):

$$IDF_t = \log \frac{N}{DF_t} \quad (2)$$

where N denotes the number of documents in the collection. In this case, rare terms that appear in some documents are given a high score. For combining the influence of both TF and IDF measures, an efficient statistical weighting method (i.e., TF-IDF) provided by Equation (3) is recommended as a term weighting scheme in information retrieval and text classification fields. Consequently, a term in a piece of text (or document) is given a high weight when frequent within a small number of documents. In contrast, it is assigned a lower score when it occurs in most documents or fewer times in a document [36]:

$$TF-IDF_t = w_t \times IDF_t \quad (3)$$

where w_t and IDF_t are calculated according to Equations (1) and (2), respectively.

3.2. Harris Hawks Optimizer (HHO)

HHO is a novel swarm intelligence (SI) meta-heuristic optimization algorithm that has been recently proposed by Heidari et al. [20]. The algorithm's idea is based on the simulation of the natural chasing styles carried out by the smart birds called Harris hawks.

The social interactions between these birds while hunting the prey (i.e., rabbit) were transformed into a novel mathematical optimization model that can be utilized for handling various optimization problems. HHO is a population-based method where each hawk plays the role of search agent while the prey is generated based on the best solution obtained so far. In alternative words, the hawks are the candidate solutions and the best candidate solution in each step is considered as the intended prey or nearly the optimum. The group of search agents collaborates utilizing different styles of exploration and exploitation strategies. The following subsections detail all phases of HHO along with the corresponding mathematical model. Table 1 summarizes the symbols employed in HHO's mathematical equations.

Table 1. Symbols of the HHO algorithm.

Symbol	Description
X	Position vector of hawks (search agents)
X_{rand}	Position of randomly selected agent
X_{prey}	Position vector of best agent so far (rabbit)
X_{avg}	Average position of all agents
$r_1, r_2, r_3, r_4, r_5, q$	uniformly random numbers within [0,1]
t, T, N	current iteration, number of iterations, number of search agents
LB, UB	Lower and upper bounds of decision variables
D	dimension
J	Random jump strength of the prey during the fleeing procedure
S	Random vector by size $1 \times D$
E_0, E	Initial value of escaping energy, escaping energy

3.2.1. Exploration Phase

In nature, Harris' hawks begin the exploration process by perching randomly at the target location and waiting to discover a rabbit. Inspiring this mechanism, two mathematical rules were proposed such that each of them has a probability of 50% of being selected as shown in Equation (4):

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)| & p \geq 0.5 \\ (X_{prey}(t) - X_{avg}(t)) - r_3(LB + r_4(UB - LB)) & p < 0.5 \end{cases} \quad (4)$$

where $X(t)$ refers to hawks' current position, $X(t+1)$ denotes hawks' position vector in the next iteration t . The average position of all agents is calculated given Equation (5):

$$X_{avg}(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (5)$$

3.2.2. Shifting from Exploration to Exploitation

To handle the problem of early convergence and avoid stuck into low-quality local optima, efficient balancing between exploration and exploitation mechanisms is employed in HHO. For this purpose, the decreasing energy of the rabbit during escaping behavior is modeled as an adaptive control parameter to move smoothly from exploration to exploitation. This dynamic parameter can be updated using the formula in Equation (7):

$$E_0 = 2r - 1 \quad (6)$$

$$E = 2E_0(1 - \frac{t}{T}) \quad (7)$$

where r is a random number inside [0,1], E_0 is randomly changed inside the interval $[-1, 1]$ at each iteration. Accordingly, E is dynamically fluctuated inside $[-2, 2]$ over the course of iterations. Figure 1 demonstrates the behavior of E during two runs and 500 iterations—in

short, when ($|E| \geq 1$) exploration equations are performed, and when ($|E| < 1$) exploitation equations are performed.

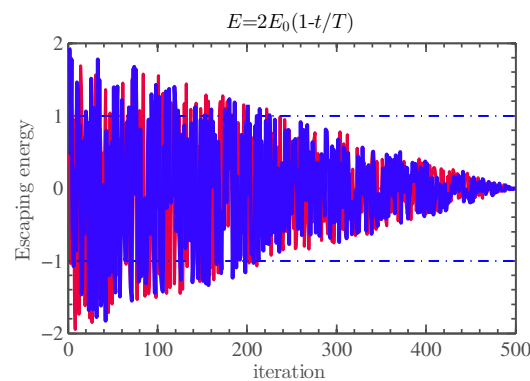


Figure 1. Behavior of E during two runs and 500 iterations [20].

3.2.3. Exploitation Phase

To exploit the neighborhood of the explored regions in the search space, four different rules are employed in the HHO, namely: soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives. The transition between these strategies is based on the escaping energy (E) and a random number r (represent the rabbit's escaping behavior). These four rules along with their mathematical formulas are summarized as follows:

- **Soft besiege:** This phase simulates the behavior of the rabbit when it still has energy and tries to escape ($r \geq 0.5$ and $|E| \geq 0.5$). During these attempts, the hawks try to encircle the prey softly. This behavior is modeled by Equations (8) and (9):

$$X(t+1) = \Delta X(t) - E|JX_{prey}(t) - X(t)| \quad (8)$$

$$\Delta X(t) = X_{prey}(t) - X(t) \quad (9)$$

where $\Delta X(t)$ refers to the difference between the prey's position and the current hawk's location in iteration t , $J = 2(1 - r_5)$ represents the random jump strength of the prey during the escaping operation. The J value changes randomly in each iteration to simulate the nature of rabbit motions [20].

- **Hard besiege:** The next level of the encircling process will be performed when the rabbit is extremely exhausted, and its escaping energy is low ($r \geq 0.5$ and $|E| < 0.5$). Accordingly, the hawks encircle the intended prey hardly. This behavior is modeled by Equation (10):

$$X(t+1) = X_{prey}(t) - E|\Delta X(t)| \quad (10)$$

- **Soft besiege with progressive rapid dives:** When the prey still has sufficient energy to escape ($|E| \geq 0.5$) but ($r < 0.5$), hawks can decide (evaluate) their next action based on the rule represented in Equation (11):

$$Y = X_{prey}(t) - E|JX_{prey}(t) - X(t)| \quad (11)$$

After that, if the possible result of such a movement is not good (the prey is performing more deceptive motions), then the hawks will start to achieve rapid and irregular dives utilizing the levy flight (LF) pattern attained by Equation (12):

$$Z = Y + S \times LF(D) \quad (12)$$

where D indicates the dimension of the given search space, S denotes a random vector with size $1 \times D$, and LF represents the levy flight function. LF -based values can be obtained using Equation (13):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \quad (13)$$

where u, v are random numbers inside $(0,1)$, β is a constant equals 1.5, and σ attained via Equation (14):

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \quad (14)$$

where $\Gamma(x)$ is the standard gamma function. Hence, the final movement in this stage can be performed by Equation (15):

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (15)$$

where $F(x)$ denotes the fitness function, Y and Z can be calculated using Equations (11) and (12).

- **Hard besiege with progressive rapid dives:** In the case of ($r < 0.5$ and $|E| < 0.5$), the hawks perform a hard besiege to approach the prey. They try to decrease the distances between their average location and the targeted prey. Accordingly, the rule given in Equation (16) is performed:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (16)$$

where Y and Z can be calculated using Equations (17) and (18):

$$Y = X_{prey}(t) - E|JX_{prey}(t) - X_{avg}(t)| \quad (17)$$

$$Z = Y + S \times LF(D) \quad (18)$$

4. The Proposed Methodology

This section presents the detailed phases of the proposed methodology designed to develop a well-performing classification model for the smart detection of fake news in Arabic tweets. As depicted in Figure 2, it starts by selecting tweets dataset collected and annotated in advance utilizing Twitter Application Programming Interface (API). Then, different NLP pre-processing techniques were applied. After that, the BoW model was used to transform tweets into numerical features. Then, extensive experiments were conducted to select the suitable ML algorithm and vectorization model. Finally, a wrapper FS method utilizing HHO and other meta-heuristics was applied to filter out irrelevant features and enhance the adopted classifier's performance.

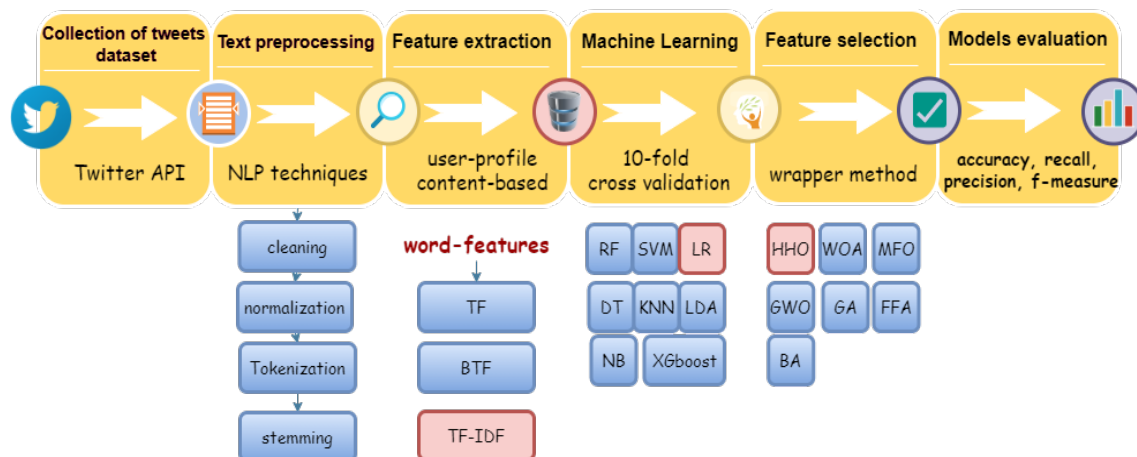


Figure 2. Conceptual diagram of the proposed methodology.

4.1. Collection of the Tweets Dataset

An Arabic Twitter corpus was utilized in this research to assess the efficiency of the proposed approach. The tweets were originally collected by [42] utilizing Twitter API. The retrieved tweets covered two main trendy topics relating to the Syrian crisis: “Syrian revolution” and “the forces of the Syrian government”. It is worth noting that these are the topics that contain the most relevant tweets [42]. The Syrian crisis tweets are appropriate to assess the credibility as many activists have relied on the Twitter platform to publish tweets that may contain misleading information.

To annotate tweets, a web-based interface has been built that allows annotators to complete their annotations. Annotators have relied on cues provided by the tweet text, author profile, and Google search on the tweet text. Based on the analyzed results, each tweet was labeled as either credible (real), non-credible (fake), or undecidable (i.e., can’t be judged to be credible or non-credible). Table 2 presents the description of retrieved tweets and the two classes’ percentage.

Table 2. Distribution of the originally collected tweets for every targeted topic.

Query Topic	No. of Tweets	Credible (%)	Non-Credible (%)	Undecidable (%)
The forces of the Syrian government	1791	63.1%	28.5%	8.4%
Syrian revolution	1232	35.6%	51.0%	13.4%

The introduced dataset by [42] is composed of 2708 tweets. In 2019, the same dataset was utilized for research by [31] in which a total of 846 tweets were discarded due to the difficulty in accessing them using the public Twitter API. Accordingly, 1862 tweets were used for credibility classification, in which the number of credible tweets (positive class) equals 1051, while the number of non-credible tweets (negative class) equals 810. In this research, we used the reduced dataset in which each tweet is described by the plain unprocessed text, and a total of 45 features extracted via the *Tweepy* (<http://docs.tweepy.org/en/v3.5.0/index.html> accessed on 26 March 2021) Python package. The list of extracted features will be presented in Section 4.3.

4.2. Data Preprocessing

Text processing is an essential step for preparing text suitable for applying mining techniques with high efficiency. In this regard, a set of NLP processing tasks were implemented. The process starts by filtering out any irrelevant text (cleaning), then normalization, tokenization, stemming, and text vectorization for feature extraction. The main processing steps of previously collected data are shown in Figure 3.

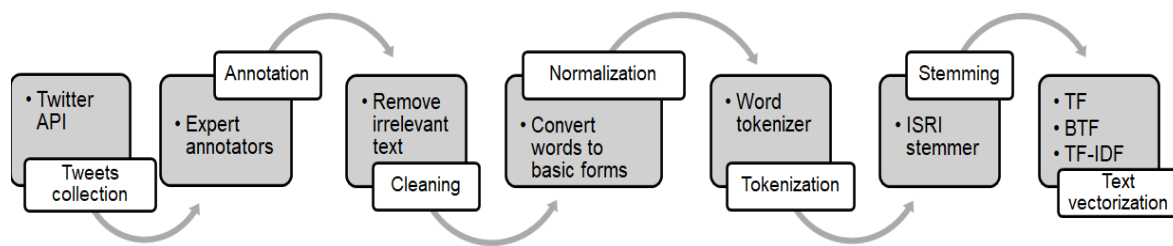


Figure 3. Process of tweets collection and preprocessing.

One of the main challenges in SA of Arabic tweets is the occurrence of colloquial words, URLs, hashtags, misspelled words, symbols, etc. Thus, the tweets are noisy and unstructured data that need preprocessing for extracting relevant text. The primary step in tackling unstructured tweets is the cleaning process. In this step, we filtered out the irrelevant text such as Arabic stop words, punctuation, numbers, non-Arabic text, web links, hashtags, and symbols.

Normalization is the following fundamental preprocessing step in which non-standard words are transformed into their basic forms. For further SA, we performed word tokenization utilizing the Natural Language Toolkit (NLTK) library [43] in the Python platform. Accordingly, tweets are split into a set of words using the white-space character as a delimiter. Another important NLP processing step is called stemming. ISRI Arabic stemmer [40] within Python NLTK was utilized in this research for further normalization of the tokenized words. ISRI is an Arabic stemmer algorithm that does not use a root dictionary. It is a widely, well-performed stemmer used in the field of Arabic tweets mining [44,45].

4.3. Feature Extraction

The goal of text vectorization techniques is to convert the unstructured tweet text into structured numerical features to apply ML algorithms for mining and knowledge extraction purposes. Generally, the retrieved Twitter data are recognized as a mixture of unstructured data (tweet text) and structured data related to content-based features and user-profile features.

The utilized dataset in this research comprises tweets' text and 45 features extracted via *Tweepy* API [31]. The extracted features were classified into two groups: content-related features and user-profile related features. The number of the content-based features is 26; some of them were retrieved directly via Twitter API (e.g., *has_mention*, *is_reply*), whereas the others were derived from the tweet text (e.g., *has_positive_sentiment*, *#_unique_words*). The remaining 18 Features related to the author's profile were retrieved directly with each tweet. The complete list of content-based and user-profile features are listed in Table 3.

Table 3. List of previously extracted features [31].

User-Profile Features		Content-Based Features		
<i>avg_hashtags</i>	<i>avg_URLs</i>	<i>has_mention</i>	<i>mentions_count</i>	<i>URL_shortner</i>
<i>followers_to_friends</i>	<i>avg_retweet</i>	<i>retweets_num</i>	<i>is_retweet</i>	<i>is_reply</i>
<i>followers_count</i>	<i>tweet_time_spacing</i>	<i>retweeted</i>	<i>day_of_week</i>	<i>#_words</i>
<i>focused_topic</i>	<i>default_image</i>	<i>length_chars</i>	<i>has_URL</i>	<i>URLs_count</i>
<i>has_desc</i>	<i>desc_len</i>	<i>has_hashtag</i>	<i>hashtags_count</i>	<i>#_unique_words</i>
<i>username_len</i>	<i>avg_tweet_len</i>	<i>#_unique_chars</i>	<i>has_?</i>	<i>#_?</i>
<i>listed_count</i>	<i>status_count</i>	<i>has_!</i>	<i>#_!</i>	<i>#_ellipses</i>
<i>retweet_fraction</i>	<i>friends_to_follower</i>	<i>#_symbols</i>	<i>has_pos_sent</i>	<i>has_neg_sent</i>
<i>is_verified</i>	<i>registration_diff</i>	<i>pos_score</i>	<i>neg_score</i>	

In addition to previously extracted features, we applied different NLP techniques to extract numerical word features from tweets text that are usable for ML algorithms. These techniques are TF, TF-IDF, and BTF that are previously explained in Section 3.1.2.

Consequently, the dataset is composed of three groups of features: word features, user-profile features, and content-based features.

To judge the effectiveness of feature categories on the performance of credibility classification, different data combinations were generated based on feature combinations. The 15 combinations of data are as follows:

- User-profile features only.
- Content-based features only.
- Word features only including either TF, TF-IDF, or BoW.
- User-profile and content-based features
- User-profile features and one set of word features (TF, TF-IDF, BoW).
- Content-based features and one set of word features (TF, TF-IDF, BoW).
- User-profile features and content-based features and one set of word features (TF, TF-IDF, BoW).

4.4. Selection of Classification Algorithms

Various state-of-the-art classification algorithms are useful for the credibility classification of tweets. However, the No Free Lunch (NFL) theorem presented for optimization [46] is valid for the classification domain [47]. That is to say, no universal classifier can efficiently handle all the problems. Accordingly, several classification algorithms were investigated, and extensive comparisons were performed to select a suitable classifier. We adopted KNN, DT, NB, LR, Linear Discriminant Analysis (LDA), SVM, RF, and eXtreme Gradient Boosting (XGboost). All classifiers were tested under the same computation environment utilizing the Python *Scikit-learn* ML library [48].

4.5. Feature Selection Using Binary HHO

The original variant of HHO is proposed to deal with continuous optimization problems and hence cannot be adapted directly to handle binary optimization problems such as FS. Therefore, the real operators of HHO should either be reformulated or boosted with additional binarization techniques without the need for operator modifications [22]. The latter is widely used in this regard, as the two-step conversion technique is applied to map the real solution into binary [23]. Firstly, fuzzy Transfer Functions (TFs) are adopted to normalize the real values into intermediate probability values within [0,1], such that each real value is given a probability of being 1 or 0. In the second step, a specific rule is applied to convert the outcome of TFs into binary.

TFs are classified based on their shape into two families: S-shaped and V-shaped [49]. In this study, we adopted four V-shaped TFs to develop a binary variant of HHO for the FS problem. The original V-shaped TF (V2) proposed by Rashedi et al. [50] in addition to the other three TFS (V1, V3, and V4) proposed in [49]. Figure 4 depicts the V-shaped TFs, and Table 4 presents their mathematical equations (where x refers to the continuous solution generated by the HHO algorithm). It is worth mentioning that these TFs were selected as they showed a clear superiority with different binary algorithms compared to S-shaped TFs [49–51].

As mentioned earlier, the resulting normalized solution from TF is converted into binary values using the binarization rule responsible for the second step of the conversion process. Two main rules, namely standard and complement, have been proposed in the literature. The standard rule (shown in Equation (19)) was initially combined with S-shaped TF by Kennedy and Eberhart [52] to design a binary variant of the PSO algorithm, whereas the complement rule (shown in Equation (20)) was initially developed by Rashedi et al. [50] to introduce a binary version of Gravitational search algorithm (GSA) algorithm. Later, this rule came to be used with V-shaped TFs. Accordingly, the complement method, along with the four V-shaped TFs, was adopted in this work.

$$X_i^j(t+1) = \begin{cases} 0 & r < T(X_i^j(t)) \\ 1 & otherwise \end{cases} \quad (19)$$

$$X_i^j(t+1) = \begin{cases} \sim X_i^j(t) & r < T(X_i^j(t)) \\ X_i^j(t) & \text{Otherwise,} \end{cases} \quad (20)$$

where t refers to the current iteration, r is a random number within $[0,1]$ using a uniform distribution, $T(x_i^j(t))$ is the probability value computed by the TF, $X_i^j(t+1)$ and $X_i^j(t)$ are the new and the current binary value for the j th dimension of the i th solution, respectively. \sim indicates the complement.

Table 4. V-shaped Transfer functions.

Name	Transfer Function Formula	Function
V1	$T(x) = \operatorname{erf}(\frac{\sqrt{\pi}}{2}x) $	
V2	$T(x) = \tanh(x) $	
V3	$T(x) = (x)/\sqrt{1+x^2} $	
V4	$T(x) = \frac{2}{\pi} \arctan(\frac{1}{2}x) $	

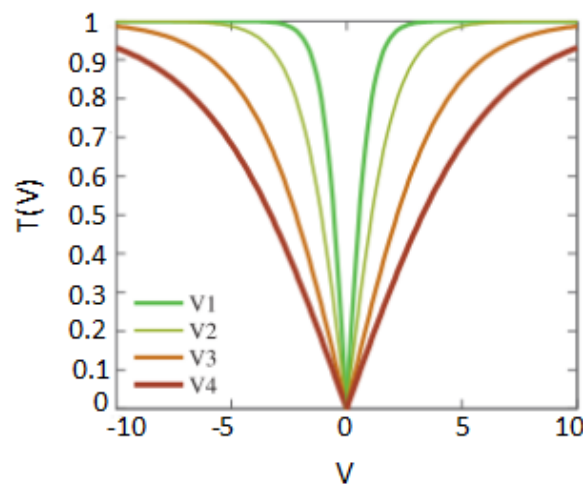


Figure 4. V-shaped Transfer functions.

Another essential aspect to consider when designing the wrapper FS approach is the evaluation (i.e., fitness) function, representing the score given for each subset of features. The desired objective of FS is to maximize the classification rate with the least number of features. For this purpose, we used the formula in Equation (21) to combine these two contradictory objectives [17,19,53]:

$$Fitness = 0.9 \times Err + 0.1 \times \frac{N_F}{D} \quad (21)$$

where Err refers to the classification error rate, N_F is the number of selected features, and D is the number of total features.

4.6. Performance Evaluation Measures

Evaluation is a significant step for measuring the performance of text mining and information retrieval approaches. The classification quality is usually estimated utilizing a special table named the confusion matrix. Table 5 depicts the used confusion matrix to assess the proposed tweets classification system in which tweets are classified as fake (negative) or real (positive). The outcomes will be one of four categories: True Positives (TP) represents the number of real tweets that are correctly predicted as real. False Positives (FP) represents the number of fake tweets incorrectly predicted as real. True Negatives (TN) refers to the number of fake tweets that are correctly predicted as fake, and False Negatives (FN) refers to the number of real tweets incorrectly predicted as fake.

Table 5. Confusion matrix.

		Predicted Class	
		positive	negative
Actual class	positive	TP	FN
	negative	FP	TN

Various evaluation measures can be derived from the outcomes of the confusion matrix. In this study, the used measures are *accuracy*, *recall*, *precision*, and *F1_measure* which are widely used in the text classification domain [2,12,31]. Details and mathematical expression of these evaluation measures are listed as follows:

- Accuracy: It is the number of correctly classified real and fake tweets divided by the total number of predictions. Accuracy is calculated using Equation (22):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (22)$$

- Recall: It is known as true positive rate, which is the ratio of correctly recognized tweets as positive (real) over the total number of actual positives, calculated as in Equation (23):

$$Recall = \frac{TP}{TP + FN} \quad (23)$$

- Precision: It is known as a positive predictive value, which is the fraction of tweets that are correctly recognized as positive (real) among the total number of positive predictions, calculated as in Equation (24):

$$Precision = \frac{TP}{TP + FP} \quad (24)$$

- F1_measure: It is defined as a harmonic mean of recall and precision metrics and can be computed as given in Equation (25). It is known as a balanced F-score since recall and precision are evenly weighted. This measure is informative in terms of incorrectly classified instances and useful to balance the prediction performances on the fake and real classes:

$$F1 - measure = 2 \cdot \frac{recall \cdot precision}{recall + precision} \quad (25)$$

5. Results and Discussion

5.1. Experimental Setup

This section describes briefly the system environment and parameters' settings employed for carrying out the experimental work. To ensure a fair comparison, all experiments were implemented under the same computing environment and coded in Python that facilitates the implementation of NLP, ML, and meta-heuristic algorithms using different libraries including *NLTK*, *Pandas*, *Numpy*, *Matplotlib*, and *SKlearn*. Table 6 presents the detailed parameter settings for all wrapper methods. To estimate the performance of machine learning algorithms, we applied 5-fold cross-validation (CV). The data sample is split into five parts, such that four parts (80%) are used for training, while the remaining part (20%) is used for testing. This process is repeated five times, thus ensuring that each sample has the chance to appear in the training and testing set. Therefore, CV yields less optimistic (or less biased) results compared to the simple hold-out method [53–55]. Due to the stochastic nature of tested meta-heuristic algorithms, all results are obtained over 10 independent runs, and average and standard deviation are reported. Moreover, Friedman's mean rank was used to rank each tested algorithm. Kindly note that the best results are highlighted using the **boldface** format.

Table 6. Common and internal settings of tested algorithms.

Common Parameters		
population size		20
Number of iterations		100
Number of runs		10
Dimension		#features
K for cross validation		5
Fitness function		alpha = 0.99 , Beta = 0.01
Internal Parameters		
Algorithm	parameter	value
BHHO	Convergence constant E	[2 0]
BWOA	convergence constant a	[2 0]
	Spiral factor b	1
BBA	Q_{min}, Q_{max}	0, 2
	loudness A	0.5
	Pulse rate r	0.5
BGWO	convergence constant a	[2 0]
BFFA	α	0.5
	β	0.2
	γ	1
BMFO	Convergence constant a	[-1 -2]
	Spiral factor b	1
GA	crossover probability	0.9
	mutation probability	0.01
	crossover type	single point
	selection strategy	roulette wheel
	elite	2
BCS	Discovery rate of alien solutions p_a	0.25

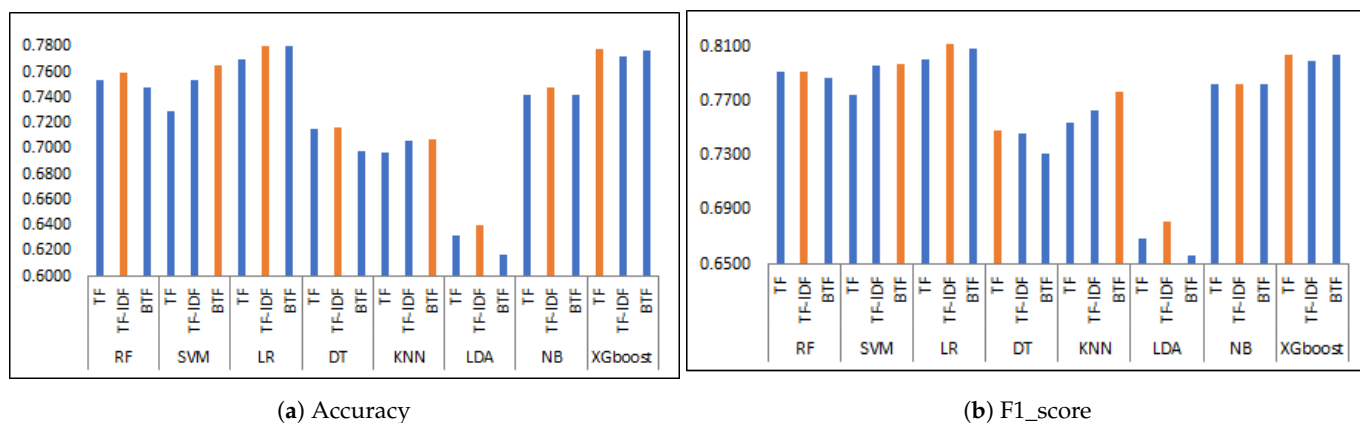
5.2. Evaluation of Classification Methods

In this paper, we examined the performance of eight different classifiers based on three different vectorization methods (i.e., TF, TF-IDF, and BTF). Alongside the evaluation criteria (i.e., accuracy, recall, precision, F1_score), Table 7 explores the obtained results. The performance of the TF-IDF model outperforms other models in four classifiers (i.e., RF, LR, DT, LDA, and NB) based on the accuracy and F1_score values. At the same time, the performance of the BTF model of word features outperforms other models in three classifiers (i.e., SVM, LR, and kNN). The performance of the TF model is better than other models only with the XGboost classifier with accuracy equal to 0.7781, and the F1_score equals 0.8042.

From the reported results, we can notice that the TF-IDF model has a better rank (i.e., 2.63) with LR classifier, while the BTF model has the worst one (i.e., 24.0) with a LDA classifier. Figure 5 explores histogram diagrams for accuracy and F1_score. The TF-IDF method's superiority can be justified because it gives a high weight for rare terms that frequently appears within a small number of tweets. These rare terms may be more relevant for the classification process. In this regard, only the TF-IDF vectorization method is adopted in all subsequent experiments.

Table 8 explores the performance of all classifiers with different combinations of features. Here, we have three types of combinations: word features, user-related, and content-related. All classifiers' performance with all combinations of features is much better than any other reported combinations (presented in Table 8). For example, the performance of LR with all features is 0.7808 and 0.8118 for accuracy and F1_score, respectively. In contrast, the performance of LR is reduced with any other combinations.

The worst performance was obtained when using user-related features only. The accuracy of all classifiers is between 0.5787 and 0.6889. Thus, it is clear that using only user-related features is not sufficient for all classifiers to build an accurate model. In this regard, the dataset with all features is considered in the rest of this work.



(a) Accuracy (b) F1_score
Figure 5. Accuracy and F1_score for all classifiers using different vectorization methods.

Table 7. Classification performance of classifiers over generated data using different vectorization methods.

Classifier	Vectorizer	Accuracy	Recall	Precision	F1_Score	Rank
RF	TF	0.7539	0.8240	0.7603	0.7909	8.63
	TF-IDF	0.7598	0.8059	0.7771	0.7912	10.25
	BTF	0.7480	0.8211	0.7544	0.7863	10.50
SVM	TF	0.7297	0.8202	0.7330	0.7741	14.75
	TF-IDF	0.7539	0.8497	0.7485	0.7959	8.13
	BTF	0.7652	0.8145	0.7796	0.7966	8.50
LR	TF	0.7695	0.8192	0.7827	0.8006	7.00
	TF-IDF	0.7802	0.8392	0.7861	0.8118	2.63
	BTF	0.7802	0.8173	0.7983	0.8077	4.63
DT	TF	0.7152	0.7507	0.7465	0.7486	17.75
	TF-IDF	0.7163	0.7364	0.7551	0.7457	16.75
	BTF	0.6980	0.7279	0.7349	0.7314	19.75
KNN	TF	0.6964	0.8221	0.6957	0.7536	16.25
	TF-IDF	0.7055	0.8382	0.6998	0.7628	14.75
	BTF	0.7077	0.9001	0.6830	0.7767	13.75
LDA	TF	0.6319	0.6584	0.6798	0.6689	23.00
	TF-IDF	0.6400	0.6803	0.6816	0.6810	22.00
	BTF	0.6169	0.6489	0.6647	0.6567	24.00
NB	TF	0.7415	0.8211	0.7465	0.7821	12.38
	TF-IDF	0.7474	0.8049	0.7615	0.7826	12.50
	BTF	0.7415	0.8211	0.7465	0.7821	12.38
XGboost	TF	0.7781	0.8069	0.8015	0.8042	5.50
	TF-IDF	0.7727	0.7992	0.7985	0.7989	7.75
	BTF	0.7770	0.8097	0.7983	0.8040	6.50

5.3. Assessing the Impact of n-Grams

In the second phase of experiments, we examined the performance of the best classifiers (i.e., LR and XGboost) based on reported results in Table 8. Here, we employed four types of n-grams: 1-gram, 2-grams, 3-grams, and 1-gram and 2-grams. Table 9 presents the obtained results for both classifiers. LR and XGboost classifiers with a 1-gram setting performing better than other n-gram settings. For example, the best performance of LR based on the F1_score value is 0.8118 with 1-gram, while the worst performance of LR obtained with 3-grams on the F1_score value is 0.7858.

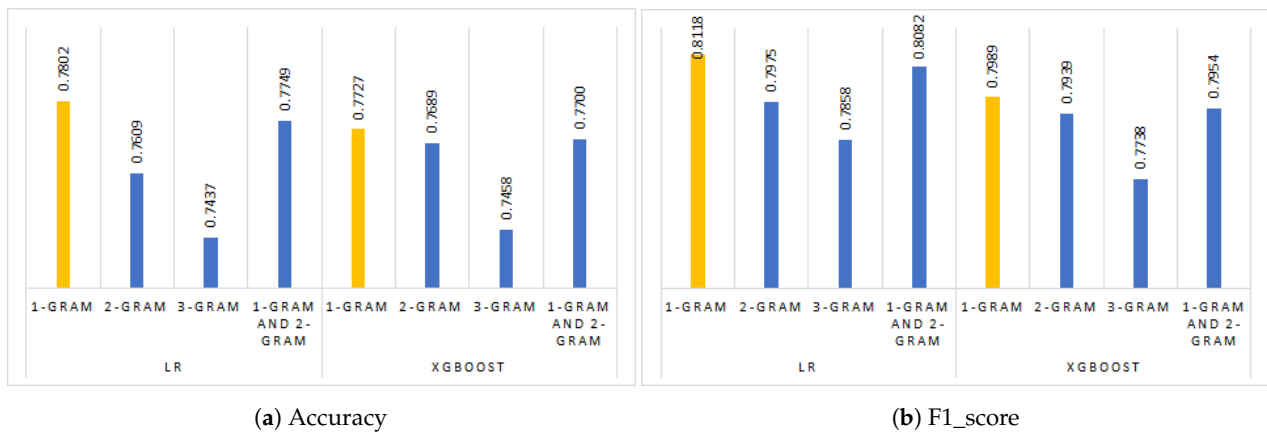
Figure 6 explores the accuracy and F1_score for both classifiers. The performance of LR classifier with 1-gram outperforms XGboost classifiers in terms of accuracy 0.7802 and F1_score 0.8118, while the performance of 3-grams either for LR and XGboost is the worst.

Table 8. Classification performance of classifiers over all combinations of features.

Word Features	User-Related	Content-Related	Classifier	Accuracy	Recall	Precision	F1_Score
✓	✓	✓	RF	0.7598	0.8059	0.7771	0.7912
			SVM	0.7539	0.8497	0.7485	0.7959
			LR	0.7802	0.8392	0.7861	0.8118
			DT	0.7163	0.7364	0.7551	0.7457
			KNN	0.7055	0.8382	0.6998	0.7628
			LDA	0.6400	0.6803	0.6816	0.6810
			NB	0.7474	0.8049	0.7615	0.7826
			XGboost	0.7727	0.7992	0.7985	0.7989
x	✓	✓	RF	0.7238	0.7545	0.7560	0.7552
			SVM	0.7093	0.8639	0.6953	0.7705
			LR	0.7082	0.8192	0.7092	0.7603
			DT	0.6711	0.7012	0.7121	0.7066
			KNN	0.6722	0.7688	0.6877	0.7260
			LDA	0.7131	0.8145	0.7163	0.7622
			NB	0.5287	0.2969	0.6933	0.4157
			XGboost	0.7303	0.7764	0.7535	0.7648
✓	x	x	RF	0.7534	0.7774	0.7841	0.7807
			SVM	0.7636	0.7840	0.7946	0.7893
			LR	0.7571	0.7926	0.7807	0.7866
			DT	0.7045	0.7412	0.7370	0.7391
			KNN	0.7179	0.8934	0.6945	0.7815
			LDA	0.6239	0.6708	0.6657	0.6682
			NB	0.7474	0.8049	0.7615	0.7826
			XGboost	0.7480	0.7631	0.7847	0.7738
x	✓	x	RF	0.6889	0.7022	0.7351	0.7182
			SVM	0.6577	0.8107	0.6605	0.7279
			LR	0.6432	0.7650	0.6585	0.7077
			DT	0.6486	0.6832	0.6910	0.6871
			KNN	0.6373	0.6993	0.6718	0.6853
			LDA	0.6464	0.7507	0.6658	0.7057
			NB	0.5787	0.3958	0.7363	0.5149
			XGboost	0.6711	0.6993	0.7129	0.7061
x	x	✓	RF	0.6932	0.7878	0.7041	0.7436
			SVM	0.6975	0.8839	0.6781	0.7675
			LR	0.7012	0.8259	0.6994	0.7574
			DT	0.6244	0.6594	0.6702	0.6647
			KNN	0.6760	0.7764	0.6892	0.7302
			LDA	0.7012	0.8316	0.6975	0.7587
			NB	0.4718	0.1979	0.5977	0.2974
			XGboost	0.6814	0.7517	0.7041	0.7271
✓	✓	x	RF	0.7614	0.7954	0.7850	0.7902
			SVM	0.7603	0.7935	0.7846	0.7890
			LR	0.7577	0.7973	0.7788	0.7880
			DT	0.6862	0.6927	0.7361	0.7137
			KNN	0.7200	0.8773	0.7017	0.7797
			LDA	0.6271	0.6784	0.6670	0.6726
			NB	0.7474	0.8049	0.7615	0.7826
			XGboost	0.7517	0.7688	0.7868	0.7777
✓	x	✓	RF	0.7555	0.7992	0.7749	0.7869
			SVM	0.7517	0.8411	0.7498	0.7928
			LR	0.7711	0.8344	0.7768	0.8046
			DT	0.7109	0.7479	0.7422	0.7450
			KNN	0.7157	0.8649	0.7014	0.7746
			LDA	0.6421	0.6813	0.6839	0.6826
			NB	0.7474	0.8049	0.7615	0.7826
			XGboost	0.7577	0.7812	0.7879	0.7845

Table 9. Comparison of top classifiers using different n-grams.

Classifier	n-Grams	Accuracy	Recall	Precision	F1_Score	Rank
LR	1-gram	0.7802	0.8392	0.7861	0.8118	2.00
	2-grams	0.7609	0.8335	0.7644	0.7975	5.00
	3-grams	0.7437	0.8325	0.7440	0.7858	6.75
XGboost	1-gram and 2-grams	0.7749	0.8402	0.7787	0.8082	2.50
	1-gram	0.7727	0.7992	0.7985	0.7989	3.50
	2-grams	0.7689	0.7878	0.8000	0.7939	4.75
	3-grams	0.7458	0.7697	0.7779	0.7738	7.25
	1-gram and 2-grams	0.7700	0.7916	0.7992	0.7954	4.25

**Figure 6.** Accuracy and F1_score for LR and XGboost using different n-grams.

5.4. Analysis of BHHO-Based Feature Selection

In data mining, data dimensionality and quality play an essential role in the overall performance of the ML classifiers. To enhance the dataset's quality, feature selection can be used to remove the redundant/irrelevant data, which enhances the quality of input data (i.e., input features). In this paper, we employed four versions of the HHO algorithm as a binary wrapper feature selection algorithm called: VBHHO1, VBHHO2, VBHHO3, and VBHHO4. The main difference between these versions is the transfer function (TF) for the binarization process, as shown in Table 4. Moreover, we employed LR as an internal classifier to evaluate the selected features by the HHO algorithm. Table 10 presents the obtained results by the four versions of HHO. Evidently, the performance of TF V-shape (V1) outperforms other versions with 0.8150, 0.8552, 0.8235, and 0.8390 for accuracy, recall, precision, and F1_score, respectively. In contrast, the performance of VBHHO4 is the worst one.

Table 10. Classification performance of BHHO and LR using V-shaped TFs.

Method	Measure	Accuracy	Recall	Precision	F1_Score	Rank
VBHHO1	Mean	0.8150	0.8552	0.8235	0.8390	1.00
	Std	0.0060	0.0058	0.0098	0.0044	
VBHHO2	Mean	0.8102	0.8540	0.8175	0.8352	2.25
	Std	0.0049	0.0151	0.0065	0.0056	
VBHHO3	Mean	0.8084	0.8441	0.8210	0.8323	2.75
	Std	0.0061	0.0127	0.0079	0.0059	
VBHHO4	Mean	0.8052	0.8432	0.8170	0.8299	4.00
	Std	0.0052	0.0070	0.0061	0.0045	

Best results are highlighted in **boldface**.

Figure 7 explores the convergences curves for four versions of BHHO. Obviously, VBHHO1 explores the search space better than other versions, while the performance of VBHHO4 is the worst. Moreover, the premature convergence for VBHHO1 takes a longer

number of iterations compared to other versions, which means that the exploration and exploitation process for VBHHO1 is much better than other versions.

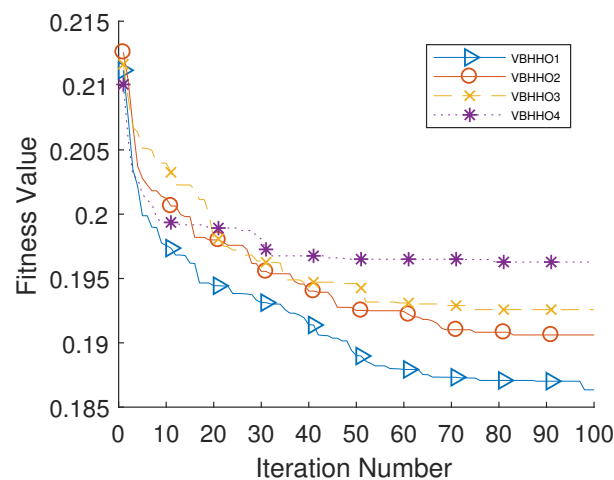


Figure 7. Convergence trends of VBHHO variants.

Figure 8 explores the histogram diagrams for the averages of selected features for four versions of VBHHO. For example, the average for VBHHO1 is 984.20 ± 246.67 . Due to the data complexity and reported results in Table 10, feature selection algorithms showed that there are redundant/irrelevant data that should be removed to enhance the overall performance of ML classifiers.

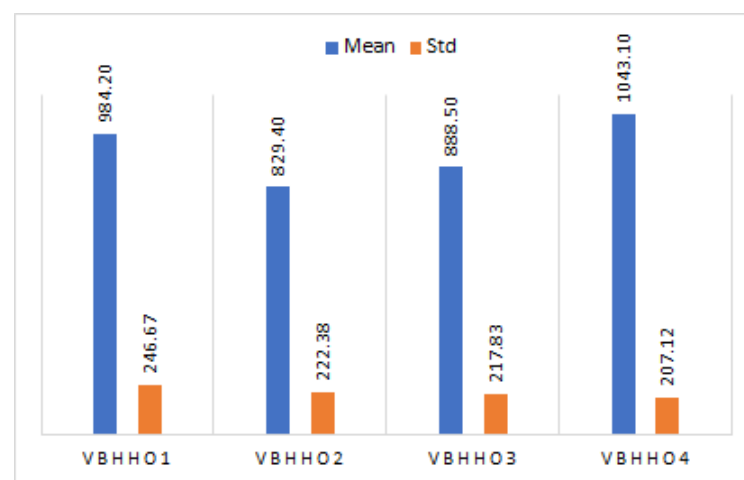


Figure 8. Average of the best-selected features by VBHHO variants.

For more data analysis, we examined seven different wrapper feature selection algorithms, which are: Binary WOA (BWOA), Binary MFO (BMFO), Binary GWO (BGWO), Binary Genetic Algorithm (BGA), Binary Firefly Algorithm (BFFA), Binary Cuckoo Search (BCS), and Binary Bat Algorithm (BBA). Table 11 explores the obtained results for these algorithms. We employed the same internal classifier (i.e., LR) and same TF (i.e., V-shape V1) for a fair comparison. Moreover, we executed each algorithm 10 times. The performance of VBHHO1 surpasses the other algorithms based on all reported criteria, while the performance of BBA is the worst one with an average accuracy of 0.7902. Figure 9 explores the average number of the best-selected features by the applied feature selection approaches (i.e., the average obtained over 10 independent runs). Successively, the VBHHO1 was able to reduce the number of features (the average equals 984.20) and recorded the best classification performance among other applied approaches. Figure 10 presents the convergence curves for all algorithms. VBHHO1 shows excellent performance compared to other algorithms.

Table 11. Classification performance of VBHHO1 and other meta-heuristics with the LR classifier.

Algorithm	Metric	Accuracy	Recall	Precision	F1_Score	Rank
VBHHO1	Mean	0.8150	0.8552	0.8235	0.8390	1.00
	Std	0.0060	0.0058	0.0098	0.0044	
BWOA	Mean	0.8045	0.8413	0.8174	0.8290	5.75
	Std	0.0038	0.0163	0.0093	0.0045	
BMFO	Mean	0.8052	0.8432	0.8171	0.8298	5.13
	Std	0.0027	0.0157	0.0067	0.0044	
BGWO	Mean	0.8073	0.8463	0.8182	0.8319	3.50
	Std	0.0025	0.0118	0.0085	0.0025	
BGA	Mean	0.8093	0.8457	0.8213	0.8333	2.50
	Std	0.0052	0.0096	0.0080	0.0045	
BFFA	Mean	0.8086	0.8432	0.8219	0.8323	3.13
	Std	0.0025	0.0132	0.0059	0.0038	
BCS	Mean	0.7936	0.8295	0.8091	0.8191	7.00
	Std	0.0056	0.0105	0.0079	0.0050	
BBA	Mean	0.7902	0.8292	0.8045	0.8166	8.00
	Std	0.0057	0.0094	0.0083	0.0049	

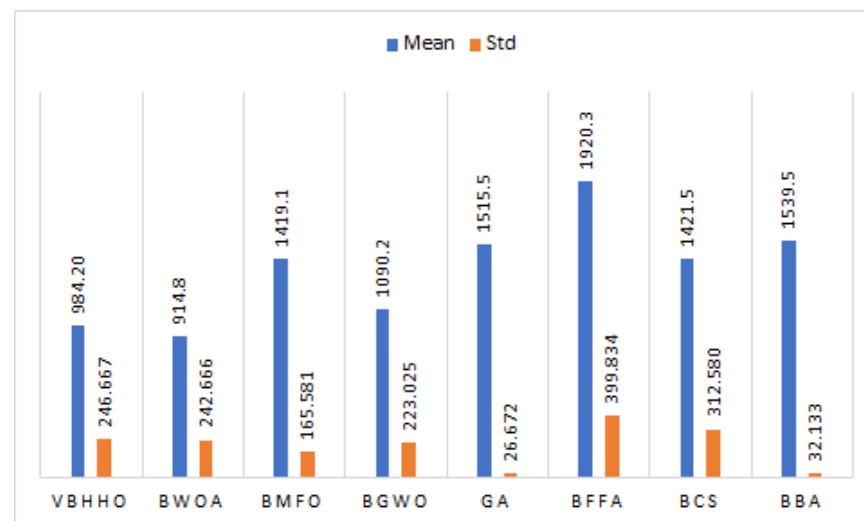


Figure 9. Average of the best-selected features by all algorithms.

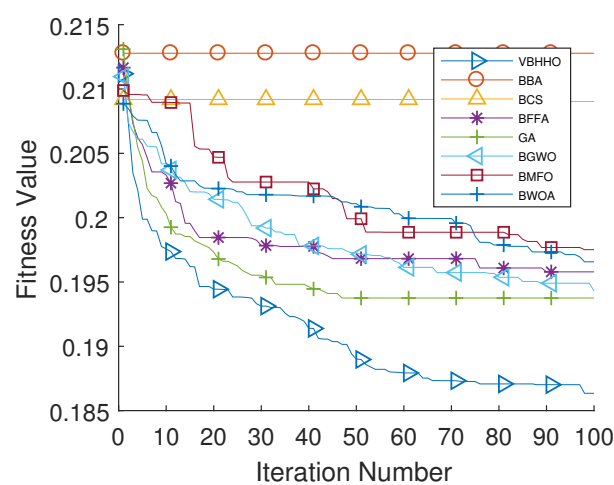


Figure 10. Convergence curves of all algorithms.

To show the impact of feature selection on LR classifier’s performance, Table 12 presents the comparison of the proposed VBHHO1-LR feature selection approach with an LR classifier without feature selection. Evidently, VBHHO1-LR provides better classification

performance with a significant reduction of features. Therefore, that proper selection of relevant features has led to performance enhancement considerably.

Table 12. Comparison of the LR classifier before and after applying VBHHO1 feature selection algorithm in terms of classification measures and number of selected features.

Method	No. Features	Accuracy	Recall	Precision	F1_Score
LR	3054	78%	84%	79%	81%
VBHHO1-LR	984.2	82%	86%	82%	84%

5.5. Comparison with Results of the Literature

To further verify the proposed approach's effectiveness, we compared the obtained results with those obtained by previous methods on the same dataset. For this purpose, we compared the results of VBHHO1-LR with RF, DT, LR, and AdaBoost proposed by Jardaneh et al. [31], and CAT proposed by Zaatari et al. [42]. From the results in Table 13, the AdaBoost algorithm surpassed other algorithms that were reported in [31] with an accuracy of 77%. The performance of the proposed VBHHO1-LR outperforms all reported results in the literature with 82%, 86%, 82%, and 84% for accuracy, recall, precision, and F1_score, respectively.

Finally, for all the reported results, it is clear that our proposed approach can enhance 5% of the overall detection performance of false information in Arabic tweets compared to other reported results in the literature, which is considered an excellent improvement in this field.

Table 13. Comparison of the proposed hybrid approach VBHHO1-LR with results from the literature.

Algorithm		Accuracy	Recall	Precision	F1_Score
proposed approach	LR	78%	84%	79%	81%
	VBHHO1-LR	82%	86%	82%	84%
Jardaneh et al. [31]	RF	76%	82%	79%	80%
	DT	70%	74%	75%	74%
	LR	76%	83%	79%	81%
	Ada Boost	77%	82%	80%	81%
Zaatari et al. [42]	CAT	-	0.791	0.79	0.788

Taken together, the comparative results proved the merits of the BHHO for tackling FS in the considered data. The superiority of HHO comes from its ability to balance between exploratory and exploitative potentials by employing the transition parameter that ensures an efficient balance between local and global search. Moreover, HHO performs well on the considered high-dimensional data due to the use of diverse exploratory mechanisms, which allow the search agents to explore more promising regions.

6. Conclusions and Future Perspectives

Detection of fake news over social media platforms is a challenging task. Nonetheless, a few studies have been presented to deal with the Arabic context over Twitter since it is a very complex language. As a result, this paper introduces a hybrid artificial intelligence model for identifying the credibility of the Arabic tweets. For this, we investigated the efficiency of most text vectorization models (TF, BTF, and TF-IDF) with several machine learning algorithms (KNN, RF, SVM, NB, LR, LDA, DT, and XGboost) to identify the efficient model that fits the tested data. The reported results revealed that LR combined with TF-IDF obtained the best performance. Moreover, this paper investigated the impact of feature selection on fake news classification. The recent HHO algorithm was adapted with the LR classifier as a wrapper FS approach. Our findings revealed that the overall quality of classification results could be further improved by filtering out the irrelevant and redundant attributes. The proposed model achieved excellent results of 0.8150, 0.8552,

0.8235, and 0.8390 for accuracy, recall, precision, and F1_score, respectively, with an increase of 5% compared to previous work on the same data.

Despite the promising results achieved in this study, the research horizon remains open to address some limitations and challenges. Automatic processing of Arabic tweets is not trivial. Most of them are written using different dialects and colloquial forms. Therefore, it will be interesting to capture the text's semantic meaning using more efficient approaches such as word embedding and compare it with the statistical vectorization methods (i.e., BoW models). In addition, it will be worth investigating the promising ensemble learning and deep learning behavior to evolve Arabic fake news prediction performance. Other favorable meta-heuristic algorithms that have been verified to be useful in other studies will also be investigated for dimensionality reduction.

Author Contributions: Conceptualization, T.T. and M.S.; Methodology, T.T. and M.S.; Data curation, T.T. and H.C.; implementation and experimental work, T.T.; Validation, T.T., M.S., H.T., and H.C.; Writing original draft preparation, T.T.; Writing review and editing, M.S., H.T. and H.C.; Proofreading, H.T.; Supervision, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: Taif University Researchers Supporting Project number (TURSP-2020/125), Taif University, Taif, Saudi Arabia.

Acknowledgments: The authors would like to acknowledge Taif University Researchers Supporting Project Number (TURSP-2020/125), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rout, J.; Choo, K.K.R.; Dash, A.; Bakshi, S.; Jena, S.; Williams, K. A model for sentiment and emotion analysis of unstructured social media text. *Electron. Commer. Res.* **2018**, *18*. [[CrossRef](#)]
2. Aljarah, I.; Habib, M.; Hijazi, N.; Faris, H.; Qaddoura, R.; Hammo, B.; Abushariah, M.; Alfawareh, M. Intelligent detection of hate speech in Arabic social network: A machine learning approach. *J. Inf. Sci.* **2020**. [[CrossRef](#)]
3. Tubishat, M.; Abushariah, M.; Idris, N.; Aljarah, I. Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Appl. Intell.* **2019**, *49*. [[CrossRef](#)]
4. Boudad, N.; Faizi, R.; Rachid, O.H.T.; Chiheb, R. Sentiment analysis in Arabic: A review of the literature. *Ain Shams Eng. J.* **2017**, *9*. [[CrossRef](#)]
5. Ajao, O.; Bhowmik, D.; Zargari, S. Sentiment Aware Fake News Detection on Online Social Networks. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 2507–2511. [[CrossRef](#)]
6. Ajao, O.; Bhowmik, D.; Zargari, S. Fake News Identification on Twitter with Hybrid CNN and RNN Models. In Proceedings of the 9th International Conference on Social Media and Society, Copenhagen, Denmark, 18–20 July 2018. [[CrossRef](#)]
7. Gupta, A.; Lamba, H.; Kumaraguru, P.; Joshi, A. Faking Sandy: Characterizing and Identifying Fake Images on Twitter during Hurricane Sandy. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 729–736. [[CrossRef](#)]
8. Castillo, C.; Mendoza, M.; Poblete, B. Information credibility on Twitter. In Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India, 28 March–1 April 2011; pp. 675–684. [[CrossRef](#)]
9. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The Rise of Social Bots. *Commun. ACM* **2014**, *59*. [[CrossRef](#)]
10. Biltawi, M.; Etaiwi, W.; Tedmori, S.; Hudaib, A.; Awajan, A. Sentiment classification techniques for Arabic language: A survey. In Proceedings of the 2016 7th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 5–7 April 2016; pp. 339–346. [[CrossRef](#)]
11. Badaro, G.; Baly, R.; Hajj, H.; El-Hajj, W.; Shaban, K.; Habash, N.; Sallab, A.; Hamdi, A. A Survey of Opinion Mining in Arabic: A Comprehensive System Perspective Covering Challenges and Advances in Tools, Resources, Models, Applications and Visualizations. *ACM Trans. Asian Lang. Inf. Process.* **2019**, *18*. [[CrossRef](#)]
12. Chantar, H.; Mafarja, M.; Alsawalqah, H.; Heidari, A.A.; Aljarah, I.; Faris, H. Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Comput. Appl.* **2020**, *32*, 12201–12220. [[CrossRef](#)]
13. Chantar, H.K.; Corne, D.W. Feature subset selection for Arabic document categorization using BPSO-KNN. In Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, 19–21 October 2011; pp. 546–551.
14. Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: New York, NY, USA, 2012; Volume 454.

15. Ahmed, S.; Mafarja, M.; Faris, H.; Aljarah, I. Feature Selection Using Salp Swarm Algorithm with Chaos. In *Proceedings of the 2nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*; ACM: New York, NY, USA, 2018; pp. 65–69. [[CrossRef](#)]
16. Zawbaa, H.M.; Emary, E.; Parv, B.; Sharawi, M. Feature selection approach based on moth-flame optimization algorithm. In *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, 24–29 July 2016; pp. 4612–4617.
17. Mafarja, M.M.; Eleyan, D.; Jaber, I.; Hammouri, A.; Mirjalili, S. Binary Dragonfly Algorithm for Feature Selection. In *Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 11–13 October 2017; pp. 12–17.
18. Zawbaa, H.M.; Emary, E.; Parv, B. Feature selection based on antlion optimization algorithm. In *Proceedings of the 2015 Third World Conference on Complex Systems (WCCS)*, Marrakech, Morocco, 23–25 November 2015; pp. 1–7.
19. Mafarja, M.; Heidari, A.A.; Habib, M.; Faris, H.; Thaher, T.; Aljarah, I. Augmented whale feature selection for IoT attacks: Structure, analysis and applications. *Future Gener. Comput. Syst.* **2020**, *112*, 18–40. [[CrossRef](#)]
20. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
21. Al-Betar, M.A.; Awadallah, M.A.; Heidari, A.A.; Chen, H.; Al-khraisat, H.; Li, C. Survival exploration strategies for Harris Hawks Optimizer. *Expert Syst. Appl.* **2020**, *114243*. [[CrossRef](#)]
22. Thaher, T.; Arman, N. Efficient Multi-Swarm Binary Harris Hawks Optimization as a Feature Selection Approach for Software Fault Prediction. In *Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 7–9 April 2020; pp. 249–254. [[CrossRef](#)]
23. Thaher, T.; Heidari, A.A.; Mafarja, M.; Dong, J.S.; Mirjalili, S.; Hawks, B.H. Optimizer for High-Dimensional, Low Sample Size Feature Selection. In *Evolutionary Machine Learning Techniques: Algorithms and Applications*; Springer: Singapore, 2020; pp. 251–272. [[CrossRef](#)]
24. Elouardighi, A.; Maghfour, M.; Hammia, H.; Aazi, F.Z. A Machine Learning Approach for Sentiment Analysis in the Standard or Dialectal Arabic Facebook Comments. In *Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco, 24–26 October 2017. [[CrossRef](#)]
25. Biltawi, M.; Al-Naymat, G.; Tedmori, S. Arabic Sentiment Classification: A Hybrid Approach. In *Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 11–13 October 2017; pp. 104–108. [[CrossRef](#)]
26. Daoud, A.S.; Sallam, A.; Wheed, M.E. Improving Arabic document clustering using K-means algorithm and Particle Swarm Optimization. In *Proceedings of the 2017 Intelligent Systems Conference (IntelliSys)*, London, UK, 7–8 September 2017; pp. 879–885. [[CrossRef](#)]
27. Al-Ayyoub, M.; Nuseir, A.; Alsmearat, K.; Jararweh, Y.; Gupta, B. Deep learning for Arabic NLP: A survey. *J. Comput. Sci.* **2018**, *26*, 522–531. [[CrossRef](#)]
28. Al-Azani, S.; El-Alfy, E.M. Combining emojis with Arabic textual features for sentiment classification. In *Proceedings of the 2018 9th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 3–5 April 2018; pp. 139–144. [[CrossRef](#)]
29. McCabe, T.J. A Complexity Measure. *IEEE Trans. Softw. Eng.* **1976**, *SE-2*, 308–320. [[CrossRef](#)]
30. Paredes-Valverde, M.; Colomo-Palacios, R.; Salas Zarate, M.; Valencia-García, R. Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach. *Sci. Program.* **2017**, *2017*, 1–6. [[CrossRef](#)]
31. Jardaneh, G.; Abdelhaq, H.; Buzz, M.; Johnson, D. Classifying Arabic Tweets Based on Credibility Using Content and User Features. In *Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Amman, Jordan, 9–11 April 2019; pp. 596–601. [[CrossRef](#)]
32. Al-Khalifa, H.; Binsultan, M. An experimental system for measuring the credibility of news content in Twitter. *IJWIS* **2011**, *7*, 130–151. [[CrossRef](#)]
33. Sabbeh, S.; Baatwah, S. Arabic news credibility on twitter: An enhanced model using hybrid features. *J. Theor. Appl. Inf. Technol.* **2018**, *96*, 2327–2338.
34. Zubiaga, A.; Liakata, M.; Procter, R. Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media. *arXiv* **2016**, arXiv:1610.07363.
35. Ruchansky, N.; Seo, S.; Liu, Y. CSI: A Hybrid Deep Model for Fake News Detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 6–10 November 2017; pp. 797–806. [[CrossRef](#)]
36. Manning, C.; Raghavan, P.; Schtextbackslashsütze, H.; Corporation, E. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 13.
37. El-Khair, I.A. Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study. *arXiv* **2017**, arXiv:1702.01925.
38. Gupta, D.; Saxena, K. Software bug prediction using object-oriented metrics. *Sadhana Acad. Proc. Eng. Sci.* **2017**, *42*, 655–669. [[CrossRef](#)]
39. Willett, P. The Porter stemming algorithm: Then, and now. *Program Electron. Libr. Inf. Syst.* **2006**, *40*. [[CrossRef](#)]

40. Taghva, K.; Elkhoury, R.; Coombs, J. Arabic stemming without a root dictionary. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), Las Vegas, NV, USA, 4–6 April 2005; Volume 1, pp. 152–157. [[CrossRef](#)]
41. Salton, G.; Wong, A.; Yang, C. A Vector Space Model for Automatic Indexing. *Commun. ACM* **1975**, *18*, 613. [[CrossRef](#)]
42. Zaatari, A.; El Ballouli, R.; Elbassuoni, S.; El-Hajj, W.; Hajj, H.; Shaban, K.; Habash, N.; Yehya, E. Arabic Corpora for Credibility Analysis. In Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016, Portorož, Slovenia, 23–28 May 2016; pp. 4396–4401.
43. Loper, E.; Bird, S. NLTK: The Natural Language Toolkit. *CoRR* **2002**. [[CrossRef](#)]
44. Faris, H.; Aljarah, I.; Habib, M.; Castillo, P.A. Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context. In Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, Valletta, Malta, 22–24 February 2020; pp. 453–460. [[CrossRef](#)]
45. Abuelenin, S.; Elmougy, S.; Naguib, E. Twitter Sentiment Analysis for Arabic Tweets. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017*; Springer International Publishing: Cham, Switzerland, 2018; pp. 467–476.
46. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
47. Larranaga, P.; Calvo, B.; Santana, R.; Bielza, C.; Galdiano, J.; Inza, I.; Lozano, J.; Armañanzas, R.; Santafé, G.; Pérez, A.; et al. Machine learning in bioinformatics. *Briefings Bioinform.* **2006**, *7*, 86–112. [[CrossRef](#)] [[PubMed](#)]
48. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
49. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [[CrossRef](#)]
50. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSa: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [[CrossRef](#)]
51. Mafarja, M.; Eleyan, D.; Abdullah, S.; Mirjalili, S. S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In Proceedings of the International Conference on Future Networks and Distributed Systems, Cambridge, UK, 19–20 July 2017; pp. 1–7.
52. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108.
53. Tumar, I.; Hassouneh, Y.; Turabieh, H.; Thaher, T. Enhanced Binary Moth Flame Optimization as a Feature Selection Algorithm to Predict Software Fault Prediction. *IEEE Access* **2020**, *8*, 8041–8055. [[CrossRef](#)]
54. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2009.
55. Thaher, T.; Mafarja, M.; Abdalhaq, B.; Chantar, H. Wrapper-based Feature Selection for Imbalanced Data using Binary Queuing Search Algorithm. In Proceedings of the 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–6. [[CrossRef](#)]