symmetry

MDPI

Article

An Evolutionary Fake News Detection Method for COVID-19 Pandemic Information

Bilal Al-Ahmad [1], Ala' M. Al-Zoubi [2,3], Ruba Abu Khurma [2] and Ibrahim Aljarah [2,*]

[1] Faculty of Information Technology and Systems, The University of Jordan, Aqaba 77110, Jordan; b.alahmad@ju.edu.jo
[2] King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan; alzoubi@correo.ugr.es (A.M.A.-Z.); ruba_abukhurma@yahoo.com (R.A.K.)
[3] School of Science, Technology and Engineering, University of Granada, 52005 Granada, Spain
* Correspondence: i.aljarah@ju.edu.jo

Abstract: As the COVID-19 pandemic rapidly spreads across the world, regrettably, misinformation and fake news related to COVID-19 have also spread remarkably. Such misinformation has confused people. To be able to detect such COVID-19 misinformation, an effective detection method should be applied to obtain more accurate information. This will help people and researchers easily differentiate between true and fake news. The objective of this research was to introduce an enhanced evolutionary detection approach to obtain better results compared with the previous approaches. The proposed approach aimed to reduce the number of symmetrical features and obtain a high accuracy after implementing three wrapper feature selections for evolutionary classifications using particle swarm optimization (PSO), the genetic algorithm (GA), and the salp swarm algorithm (SSA). The experiments were conducted on one of the popular datasets called the Koirala dataset. Based on the obtained prediction results, the proposed model revealed an optimistic and superior predictability performance with a high accuracy (75.4%) and reduced the number of features to 303. In addition, by comparison with other state-of-the-art classifiers, our results showed that the proposed detection method with the genetic algorithm model outperformed other classifiers in the accuracy.

Keywords: fake news; COVID-19; misinformation; evolutionary algorithms; metaheuristics; genetic algorithm

Citation: Al-Ahmad, B.; Al-Zoubi, A.M.; Abu Khurma, R.; Aljarah, I. An Evolutionary Fake News Detection Method for COVID-19 Pandemic Information. *Symmetry* **2021**, *13*, 1091. https://doi.org/10.3390/sym13061091

Academic Editor: Jan Awrejcewicz

Received: 9 June 2021
Accepted: 18 June 2021
Published: 20 June 2021

1. Introduction

The COVID-19 pandemic is the most critical health crisis in the whole world, affecting all aspects of life [1]. At this time, with the outbreak of the COVID-19 pandemic, social media have become widely used to obtain information about the pandemic. There is a massive amount of false information and fake news, which results in people being confused and raises the need for accurate and real information about the pandemic. Social media such as Facebook, Twitter, and Instagram have facilitated the interaction among people across the world. Thus, because of the huge amount of information and its instant spread, some of this information is real and some is fake. Fake news influences people and leads them in the wrong directions. Therefore, there is an urgent need to find an effective approach that can detect fake news about the COVID-19 pandemic.

Distinguishing fake news is not an easy task since it purposely aims to identify false information. Research on fake news detection has used different classification models. Fake news on social media can have significant negative social effects. As a result, the discovery of fake news on social platforms has attracted researchers' interest. The existing techniques for fake news detection use both news and social media content as the sources for the learning process. Fake news plays a major role in misleading people and spreading false information.

Several research studies [2–4] used various classification algorithms to detect misinformation related to the COVID-19 pandemic. The study [2] used BERT embedding and a shallow neural network to classify COVID-19 tweets. Another study [3] used ten machine learning algorithms with seven feature extraction methods to classify fake news on COVID-19. Furthermore, the study [4] used four machine learning classifiers, decision trees, logistic regression, gradient boost, and support vector machine, to detect fake news on social media.

With the advent of deep learning, there has been a great development in the field of text classification, and thereby in fake news classification. The study [5] used convolutional neural networks (CNNs), long short-term memory (LSTM), and bidirectional encoder representations from transformers (BERT) to detect fake news on COVID-19. Furthermore, the study [6] developed an approach based on an ensemble of three transformer models (BERT, ALBERT, and XLNET) to detect fake news. The model was trained and evaluated in the context of the Constraint AI 2021 Fake News Detection dataset.

The motivation of this study was to use an evolutionary fake news detection technique to extract the most important features of fake news information. Because of the number of symmetrical features related to the COVID-19 pandemic information is large, the proposed approach aimed to reduce the number of symmetrical features and obtain a high classification accuracy simultaneously after implementing four evolutionary classifications, namely k-NN-BSSA, k-NN-BPSO, k-NN-BGA, and k-NN, and three wrapper feature selection techniques (BGA, BPSO, and BSSA), with k-nearest neighbors (kNNs) as the main classifier to evaluate the output features. The fake news detection involved two primary sources of information: news websites and social media platforms. The experiments were conducted on a fake news website-based dataset called the Koirala dataset, then six datasets were constructed using different text tokenization forms (binary, TF, TF-IDF, and bag-of-words) and stemming techniques, namely Data 1, Data 2, Data 3, Data 4, Data 5, and Data 6.

The remainder the paper is organized as follows: A brief summary of the relevant studies is given in Section 2. Section 3 presents the materials and methods that were used in this research. The details of the proposed methodology are described in Section 4. Section 5 presents and discusses the results. The findings of this research are concluded in Section 6. The abbreviations used in this paper are given at the end.

2. Related Work

Social networks have an important role in our daily lives as anyone can publish their ideas and spread information without verifying the authenticity of the content. Several studies involved the discovery of fake news in the context of social media. The study [7] proposed an approach to detect fake news sites to help users avoid such fake news. The study used some features of the news to detect fake information such as keywords and punctuation marks. Furthermore, the study [8] presented different approaches to distinguish between true and fake news, with high performance.

The research paper [9] developed an approach for classifying fake users and fake news on Twitter by implementing entity recognition and hashtag, emoji, and text sentiment analysis. The study by [10] proposed deep learning models based on a feed-forward neural network (FNN) and LSTM in conjunction with different word vector representations. The proposed model mainly focused on collecting information from the news article based on the content and title. The developed model obtained good results in terms of the evaluation measures used.

Additionally, another work [11] used Twitter's data to determine the most important features that affected the performance of the machine learning methods when used for the classification of fake news. The study presented a set of new Twitter properties that could improve the classification of fake and real tweets. Furthermore, the work by [12] proposed a graph-based semi-supervised learning model to capture fake users on Twitter using certain valuable features.

Similarly, the researchers in [13] provided a text-based approach to fake news detection by using a two-layer classification. The first layer was used for detecting the fake topic, and the second layer was used for detecting the fake event. The research article [14] proposed a hybrid approach for classifying fake news on social platforms, which combined naive Bayes, support vector machines, and semantic analysis.

The study [15] introduced a two-phase technique for identifying fake news on social media. The first phase involved converting unstructured datasets into structured datasets. The second phase involved applying twenty-three supervised AI models on the BuzzFeed Political News dataset. The experimental results indicated that the J48 algorithm achieved the highest accuracy compared with the other AI models. Further, the authors in [16] described a deep learning model on a Kaggle fake news dataset. They performed text preprocessing by using word embedding (GloVe) to construct a vector space of words and created a linguistic relationship. For the classification, the authors proposed a new classification model based on two types of neural networks: convolutional and recurrent architectures. Another paper [17] captured the problem of malicious rumors that appear during breaking news. The study proposed a rumor-based propagation model called the HISBM model, which detected the propagation process of multiple rumors on online social networks to reduce the number of malicious information over a short period. Likewise, The spread of fake health-related news has a bad effect on people's sentiments. The work introduced by [18] proposed a new approach to analyze Reddit, Facebook, and Twitter content. The results indicated that news focused on fake health information was often considered as bad messaging and that news based on the evidence of social influence was acceptable and respected. Furthermore, the study [19] proposed a theory-based model to early distinguish fake news. Based on the news content, the model detected fake news when it was published on news sites before spreading on social media platforms. The results were obtained by applying the model on two real-world datasets. In the same context, the research study [20] proposed a graph-based approach for unsupervised fake news detection over multiple datasets. The proposed approach employed different graph-based techniques, such as label spreading, a graph-based feature vector, and bi-clique identification. Particular to the AAAI-2021 COVID-19 dataset, the study [21] used both machine and deep learning techniques including the SVM, CNN, BiLSTM, and CNN+BiLSTM techniques with the TF-IDF and Word2Vec embedding techniques. Additionally, the research work [22] applied sentiment analysis to identify misinformation in tweets on the COVID-19 Twitter discourse. They captured unreliable and misleading content based on fact-checking websites and investigated the narratives of misinformation tweets.

The aforementioned studies above did not introduce an evolutionary classification technique to show the most essential features of the COVID-19 fake news information. As a result, this study proposed an evolutionary-based detection approach to capture the most important features of fake news information. Our approach aimed to reduce the number of features and achieved a high accuracy. Four evolutionary classifications techniques (k-NN-BSSA, k-NN-BPSO, k-NN-BGA, and k-NN) were used. Furthermore, three wrapper feature selection techniques (BGA, BPSO, and BSSA) were implemented to evaluate the output features.

3. Materials and Methods

3.1. Feature Selection

Feature selection (FS) is a process that is performed before implementing data mining tasks such as classification and clustering [23–25]. The primary function of FS is to prepare the data (preprocess) by reducing the symmetry in the feature space (dimensionality) of a dataset [26]. FS performs a dimensionality reduction by eliminating noisy symmetrical features and keeps the most informative features in a dataset. The noisy features include features that have a high correlation with other symmetrical features (redundant) and features that have a weak correlation with the target class (label of the instance) (irrelevant) [27]. The major advantage of applying FS is the construction of a reduced version of

the dataset, which reduces the cost of the learning process in terms of time and hardware resources. FS increases the performance of the learning process because it can alleviate the problem of overfitting. The primary processes of FS are searching and evaluation [28]. The search involves a search algorithm that traverses the search space to find the global best feature subset. There are several types of search algorithms that have been adopted in the literature, such as brute-force methods and metaheuristic algorithms. The searching algorithms differ in their time complexity. Brute-force methods traverse all the generated feature subsets to decide the optimal feature subsets so that they consume exponential time to end the search process. Metaheuristic search methods generate random feature subsets and examine them until reaching the near-optimal features subset. Many of these algorithms have been enhanced to improve the FS process by adopting new operators [29], novel update strategies [30], new initialization [31], new encoding schemes [32], and new fitness functions [33], as well as applying multi-objective [34] and parallel algorithms [35]. The evaluation process of FS is accomplished based on the characteristics of the dataset (e.g., filters) or based on a learning algorithm (e.g., wrappers). Filters are fast methods because they do not involve a learning process [36,37]. However, wrappers generate more accurate results because the classifier used in the learning process of FS is normally used for the evaluation in the external testing process [30].

3.2. Nature-Inspired Algorithms

Nature-inspired algorithms (NIAs) are a set of methodologies that play a significant role in tackling various optimization problems. NIAs result from the relation of nature with different scientific fields including physics, chemistry, biology, mathematics, and engineering. Computer science has used these relations between science and nature and turned them into a well-defined discipline for optimizing different complex and challenging problems. NIAs are divided into two primary subcategories: evolutionary algorithms (EAs) and swarm-based algorithms (SI).

In EAs, the algorithms approximate biological evolution in their emulation. They include different computational systems that model natural evolutionary processes, such as reproduction, mutation, recombination, and selection. Examples of these types of algorithms are genetic algorithms (GAs) [38,39], differential evolution (DE) [40], and the biogeography-based optimization algorithm (BBO) [41].

In the SI category, the algorithms have a common behavior that is very similar to the collective (social) behavior of individuals (agents). A swarm system is comprised of an abundant number of solutions (agents) that are distributed in the environment to achieve a common (global) target. Intelligence can be seen in the swarm behavior, but not by looking at a single agent alone. Usually, swarm systems have interesting features and proprieties that help them coexist. Similarly, SIs have advantages that contribute to their success, such as adaptability, self-organization, distributed control, scalability, and flexibility. Briefly, a swarm is a distributed system where the agents' behavior is autonomously controlled with no external management. Agents in a swarm are self-organized, which implies they are not pre-defined, but updated continuously at runtime. In addition, adaptability requires that agents adapt themselves to environmental changes and update their behavior accordingly. The update and modification steps depend on several heuristics inferred from the swarm. The scalability of SI means that it can perfectly cope with an increasing the number of agents without changing the control architecture of the algorithm. This category contains a considerable number of algorithms such as particle swarm optimization (PSO) [42], the moth–flame optimization (MFO) algorithm [26], and salp swarm optimization (SSA) [29,43]. SI algorithms are characterized by two embedded conflicting milestones: exploration and exploitation. In exploration, the candidate solutions change continuously, which leads to exploring more areas and finding different solutions. In exploitation, the candidate solutions change less. Exploitation focuses on a certain promising region found by exploration and performs the search in a local region to improve the quality of the solutions.

3.2.1. Genetic Algorithm

The GA is undoubtedly the most widespread and typical example of EAs. It was first proposed by John Holland in 1975. The GA adopts Darwin's theory of natural selection and evolution. Since then, it has been widely applied in different disciplines. The GA is typically designed by identifying a set of candidate solutions called chromosomes. The set of these chromosomes builds a population. Each chromosome is divided into smaller units called genes. The length of the chromosome (the number of genes) determines the dimensionality of a problem. At each iteration of the GA, a set of evolutionary operators (selection, crossover, mutation) is applied to create diversity in the current population in order to prepare a new population in a way that simulates the natural evolution. Each chromosome is evaluated according to certain evaluation criteria (object functions) to determine its quality and decide if it is fit or unfit. The highest evaluated solution (best individual) is preserved in each iteration. The unfit solutions (worst individuals) are candidates to be replaced by the newly generated offspring. This allows the average fitness value to increase dramatically throughout the iterations. Figure 1 shows the GA's evolutionary process.

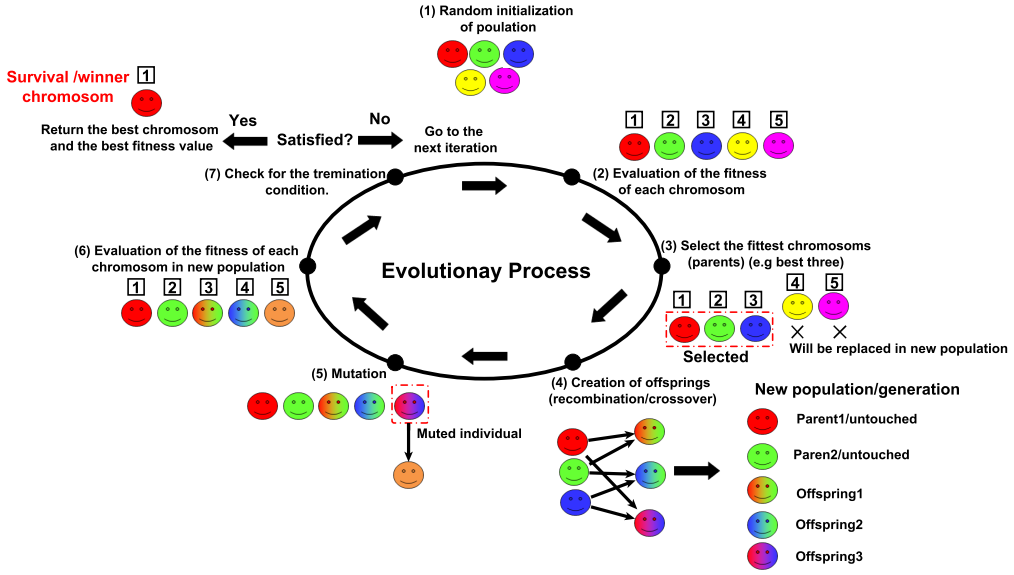


Figure 1. GA evolutionary process.

3.2.2. Binary Particle Swarm Optimization

PSO is a well-known SI algorithm that has been applied in many applications. The source of inspiration was flocks of birds searching for food. The solutions in PSO are called particles. The search procedure is guided by two primary factors: pbest and gbest. pbest is a solution that represents the best experience that was gained by the previous particle itself. gbest is the solution that represents the best solution in the entire swarm. Particles also have a position and velocity, which are both updated in each iteration based on a predefined mathematical model.

The original PSO algorithm was designed to optimize in continuous search spaces. However, the feature space is discrete. This is because FS is a binary problem in which the solutions have only two values, either "0" or "1": "1" means that the corresponding feature is selected, whereas "0" means that the corresponding feature is not selected. Hence, there is a need to perform some modifications on the continuous PSO to generate a binary version of it. There is an important point to consider, which is the position update strategy of the optimizer. The focus in continuous optimization is to keep the updated components within the upper and lower limits. However, in binary optimization, the restriction is to keep the components binary. The position update strategy in binary optimization is to switch between "0" and "1". This depends on the method used to define when the

component of the solution has to be assigned to "0" or "1". To perform this, there should be a method to convert the continuous variables into binary variables. Transfer functions are widely used methods that generate binary versions of continuous optimizers. In [44], Mirjalili and Lewis used these transfer functions to generate a binary version of the PSO algorithm. Their function defined the probability of updating a component of a solution. In [45], Kennedy and Eberhart used the sigmoid function to produce a binary version of the PSO algorithm as in Equation (1), where $v_i^d(t)$ shows the velocity of solution i for dimension d in iteration t. By applying the sigmoid function, it converts the velocity values to probability values in the range [0, 1].

$$T(v_i^d(t)) = 1/(1 + e^{-v_i^d(t)}) \tag{1}$$

The S-shaped TFs are shown in Equation (2), where $X_i^d(t+1)$ represents the ith component in the X solution for dimension d in iteration $t+1$, rand $\in [0,1]$, which is generated using a random probability distribution. It is used to update the components of the position vector after defining a probability for each of them.

$$X_i^d(t+1) = \begin{cases} 0, & \text{if } rand < S\_TF(v_i^d(t+1)) \\ 1, & \text{if } rand \geqslant S\_TF(v_i^d(t+1)) \end{cases} \tag{2}$$

Using V-shaped TFs, Equation (4) is used to update the component of the next iteration based on the probability values from Equation (3). This equation was used in [46] to convert the GSA to binary.

$$T(X_i^d(t)) = |tanh(X_i^d(t))| \tag{3}$$

$$X_{t+1} = \begin{cases} X_t, & \text{if } rand < V\_TF(\Delta X_{t+1}) \\ \neg X_t, & \text{if } rand \geqslant V\_TF(\Delta X_{t+1}) \end{cases} \tag{4}$$

3.2.3. Binary Salp Swarm Algorithm

SSA is one of the widely used SI algorithms that was inspired by a kind of animal that lives in the seas and oceans, called salps. Their behavior that is modeled is reaching a food source. The leader of the salps is the first salp in the chain. It leads the remaining salps in the chain towards the food source. The remaining salps in the chain follow the leader salp. Salps implement a dynamic movement with respect to each other. Therefore, there is a direct or indirect change in the positions of follower salps towards the leader salp. The original SSA algorithm was continuous. Therefore, the same steps followed for converting the PSO to binary were used with the SSA to convert it to binary. Figure 2 shows a single salp and the salps chain.

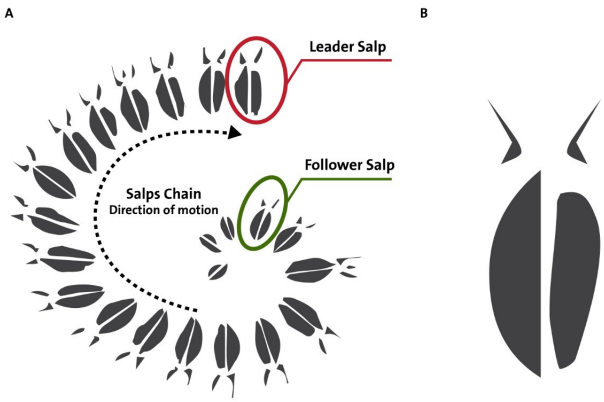


Figure 2. (**A**) Salps chain. (**B**) Single salp.

4. Methodology

The proposed methodology that was utilized in this paper consisted of five stages, namely data collection, data preprocessing and feature extraction, model development, and evaluation and assessment. All the proposed processes of the methodology can be found in Figure 3.

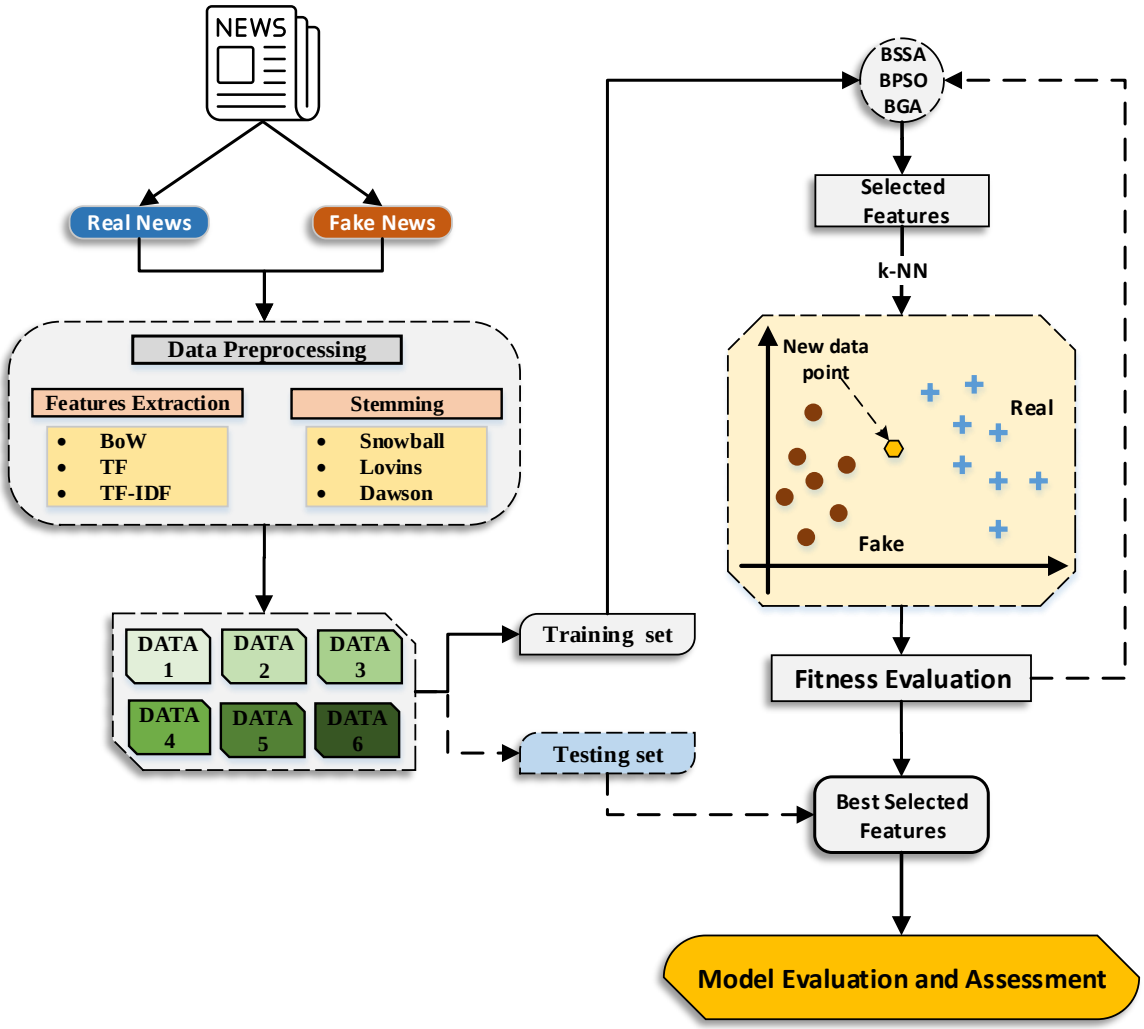


Figure 3. Methodology's processes.

4.1. Dataset Collection

One of the recent datasets in the fake news domain is the COVID-19 Fake News Dataset (https://data.mendeley.com/datasets/zwfdmp5syg/1, accessed on 20 May 2021), which was published by Abhishek Koirala [47] on the Mendeley portal. This dataset consists of a collection of true and fake news related to COVID-19. The initial dataset contained more than 6000 articles published across various media around the world and collected between December 2019 and July 2020. The news articles were collected based on three keyword searches: "corona", "coronavirus", and "COVID-19". Furthermore, the dataset was collected using the web provider Webhose.io and contained 3 subcategories of news: false news, true news, and partially false news. The data were labeled as 0 for both partially false news and false news, and true news was labeled as 1.

4.2. Data Preprocessing and Feature Extraction

This phase included several cleaning steps to eliminate the undesired parts of the data. First was the feature extraction, which was performed by tokenizing the extracted tweets. In this step, the extracted documents and text were transformed into rich practical texts to improve the classification output. The results from this step were a set of words, phrases, and sentences that were considered machine-consumable forms of text. Three models (BoW, TF, and TF-IDF) were used to extract the word features [48,49].

After the tokenization step, the following methods were applied:

- The removal of stop words: These were the highly repeated words in the text such as "the", "and", "but", "or", etc. Eliminating these words from the text reduced the dimensionality of the data, and helped build a more robust and efficient classification model [50]. This step involved eliminating some common and exclusive words;
- Stemming: This step included using different stemmers such as snowball, Lovins, and Dawson [51] to convert all the derivative words back to their basic roots. This approach was very beneficial to reduce the data dimensionality and help detect similar words in different forms;
- Eliminating unnecessary characters: These included extra spaces, punctuation marks, and other symbols [52] used in the dataset that were meaningless and unnecessary for our data analysis. Removing such unwanted characters improved the performance of the evolutionary classifier;
- Feature extraction: This paper used three popular feature extraction techniques to detect the most essential features of the COVID-19 pandemic information. Bag-of-words (BoW), term frequency (TF), and term frequency-inverse document frequency (TF-IDF)) are described as follows:
  - Bag-of-words: This is a textual approach used in multiple applications such as the machine learning classifiers [53]. Each type of text, such as sentence, paragraph, article, or document, is treated as a group of words regardless of the syntactical or semantic dependency. It presents the existence of words within a particular text;
  - Term frequency: Similar to the BoW method, term frequency is one of most popular approaches used in textual manipulation. It represents the frequency of a specific term in the correlated part of the text. The frequency is computed by finding the number of keyword occurrences in a document divided by the total number of keywords in the whole document;
  - Term frequency-inverse document frequency: Since term frequency represents the most frequent words within a document that have largest weights, some of these words are insignificant, such as the word "the". Therefore, one of the advanced methods to resolve this issue is to use the TF-IDF approach, which grants the rare words more weight than the common ones in all documents [54]. TF in the TF-IDF approach is the frequency of a particular word in the current document, whereas IDF assesses how rare the keywords are across all documents.

Six combinations of datasets were created using different text representations and stemming techniques. Table 1 shows the extracted datasets.

Table 1. The description of the datasets.

Datasets	Tokenization	Stemming	# of Features
Data 1	Binary	No	1231
Data 2	TF-IDF	No	1231
Data 3	TF	No	1231
Data 4	TF-IDF	Snowball stemmer	1240
Data 5	TF-IDF	Lovins stemmer	1223
Data 6	Bag-of-words	Dawson stemmer	611

4.3. Models Development

Three wrapper feature selections were used as the search algorithms (BGA, BPSO, and BSSA) and k-nearest neighbors (kNNs) as the main classifier to evaluate the output features. In this work, the output solution (feature subset) was represented as a binary vector of length n, where n represents the number of the features in the given dataset. If a feature was set to 1, this means that it was selected, otherwise it was not. The quality of a feature subset was determined according to the classification model accuracy and the number of selected features. Fitness is represented in the following equation:

$$Fitness = \alpha \times (1 - accuracy) + (1 - \alpha * \frac{|S|}{|W|}) \quad (5)$$

where the accuracy is the ratio of correctly classified fake and non-fake news instances over all the number of classified instances that were correctly and incorrectly classified; |S| is the number of selected features by the search algorithm, and |W| is the number of all features in the dataset; $\alpha \in [0, 1]$. All dataset combinations were divided into training and testing based on 5-fold cross-validation.

4.4. Evaluation and Assessment

Our study used four evaluation measures, which were accuracy, precision, recall, and g-mean. These measures were calculated using a confusion matrix, where the true positives (TPs) represented the news that was predicted as fake and was fake news, the true negatives (TNs) were the news that was predicted as not fake and was non-fake, the false positives (FPs) represent the news that was predicted as fake, but was non-fake, and the false negatives (FNs) were the news that was predicted as not fake, but was fake. The metrics used were defined as follows:
Accuracy represents the ratio of correctly classified fake and non-fake news instances over all the correct and incorrect classified instances. Equation (6) is the equation for accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}; \quad (6)$$

Precision, which is the ratio of news correctly identified as fake over all fake (positive) news instances, it is represented as in Equation (7).

$$Precision = \frac{TP}{FP + TP}; \quad (7)$$

Recall, which is the sensitivity of the model, means how well the model can identify the positive examples (fake) of news, which is defined by Equation (8).

$$Recall = \frac{TP}{FN + TP}; \quad (8)$$

The F-measure can be measured by calculating the weighted average of the recall and precision measures, as given by Equation (9).

$$F - Measure = \frac{(2 \times Precision \times Recall)}{(Precision + Recall).} \quad (9)$$

5. Results and Discussion

In this section, three different experimental phases are presented. The first phase described the performance of the four versions of the k-NN classification model, including k-NN-BSSA, k-NN-BPSO, k-NN-BGA, and standard k-NN. Further, the models were combined with various metaheuristic algorithms for the feature selection problem. Second, the best model was transferred to the next phase to compare with other classification algorithms.

All experiments were examined on the aforementioned datasets, namely Data 1, Data 2, Data 3, Data 4, and Data 5, each of which had a different representation and stemming technique. The splitting method applied in this work followed the five-fold cross-validation criteria. The settings of the metaheuristic algorithms are shown in Table 2.

Table 2. The settings of the parameters.

Algorithm	Parameter	Value
BSSA	c_1	[0–1]
	c_2	[0–1]
BPSO	Acceleration constants	[2.1, 2.1]
	Inertia w	[0.9, 0.6]
BGA	Single-point crossover	0.9
	Mutation	0.01

5.1. Comparisons with Other Metaheuristics

In this subsection, the comparison of various metaheuristic algorithms and the standard k-NN against five different datasets is introduced. Furthermore, this was performed to show the performance improvement by employing metaheuristic algorithms on the k-NN classifier.

As can be seen in Table 3, the k-NN-BGA obtained the highest results in terms of accuracy for Data 1 with 0.73% and 631 selected features, followed by k-NN-BSSA, k-NN-BPSO, and the basic k-NN with 0.726%, 0.725%, and 0.70%, respectively. In terms of precision, the best results were achieved by k-NN-BGA, and the second-best algorithm was k-NN-BPSO. k-NN-BPSO outperforms the other algorithms in terms of recall and the F-measure.

Table 3. Comparison of the classification results for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset with the BoW representation (Data 1), the best result is marked in bold font.

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.7264	0.6045	0.5605	0.5817	790.4
k-NN-BPSO	0.7258	0.6194	0.6030	0.6111	619.8
k-NN-BGA	**0.7348**	0.6468	0.5142	0.5729	631.2
k-NN	0.7053	0.5628	0.5888	0.5755	All

Table 4 illustrates the results of the classification methods for Data 2. In contrast to the previous experiment, the basic k-NN gained the best results in terms of accuracy, followed by k-NN-BGA, k-NN-BPSO, and k-NN-BSSA, respectively. As for the precision measure, k-NN-BGA provided the fittest results from all other algorithms with 0.64%. In terms of recall and the F-measure, k-NN-BSSA and k-NN-BPSO obtained the best results, respectively.

Table 4. Comparison of the classification results for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset with the TF-IDF representation (Data 2).

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.6161	0.4587	0.7297	0.5633	753.6
k-NN-BPSO	0.6639	0.6194	0.6030	0.6111	613.8
k-NN-BGA	0.6764	0.6468	0.5142	0.5729	619.2
k-NN	**0.7053**	0.5628	0.5888	0.5755	All

As for dataset Data 3, both k-NN-BGA and k-NN-BPSO exceeded the other algorithms based on the accuracy measure with 0.734%, as shown in Table 5. k-NN-BGA obtained the highest results in terms of precision with 0.64%. For recall and the F-measure, k-NN-BSSA achieved the best results with 0.59% and 0.60%, respectively.

Table 5. Comparison of the classification results for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset with the TF representation (Data 3).

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.7332	0.6095	0.5945	0.6019	794
k-NN-BPSO	**0.7348**	0.6039	0.5577	0.5799	613.4
k-NN-BGA	**0.7348**	0.6442	0.4877	0.5551	606.4
k-NN	0.7053	0.5628	0.5888	0.5755	All

According to the accuracy results for Data 4 in Table 6, k-NN-BGA again achieved first place with 0.75%, whereas k-NN-BSSA placed second. Further, k-NN-BGA placed first in terms of precision, and the standard k-NN was the first in terms of the recall measure. As for the F-measure, k-NN-BSSA accomplished the best results with 0.61%.

Table 6. Comparison of the classification results for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset with the TF-IDF and snowball stemming representation (Data 4).

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.7220	0.5782	0.6673	0.6196	808.6
k-NN-BPSO	0.7187	0.5810	0.6134	0.5968	625.8
k-NN-BGA	**0.7251**	0.5949	0.5955	0.5952	602.4
k-NN	0.6963	0.5408	0.6957	0.6085	All

Table 7 describes the results of the four methods for Data 5. For this dataset, k-NN-BGA achieved the highest accuracy results for all datasets, not only Data 5, with 0.75% and 602 selected features. Furthermore, k-NN-BGA obtained the best results in both precision and the F-measure. k-NN-BSSA achieved the best results in terms of recall with 0.597%.

Table 7. Comparison of the classification results for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset with the TF-IDF and Lovins stemming representation (Data 5).

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.7261	0.5962	0.5974	0.5968	789.4
k-NN-BPSO	0.7312	0.6198	0.5378	0.5759	616.2
k-NN-BGA	**0.7543**	0.6622	0.5633	0.6088	605.4
k-NN	0.7065	0.5642	0.5936	0.5785	All

Finally, the results of Data 6 are summarized in Table 8. Again, k-NN-BGA outperformed the other methods in terms of accuracy, followed by k-NN-BPSO, k-NN-BSSA, and k-NN, respectively. Similar to the accuracy results, k-NN-BGA exceeded the other methods in terms of the precision measure. As for recall and the F-measure, k-NN and k-NN-BSSA achieved the best results with 0.61% and 0.60%, respectively.

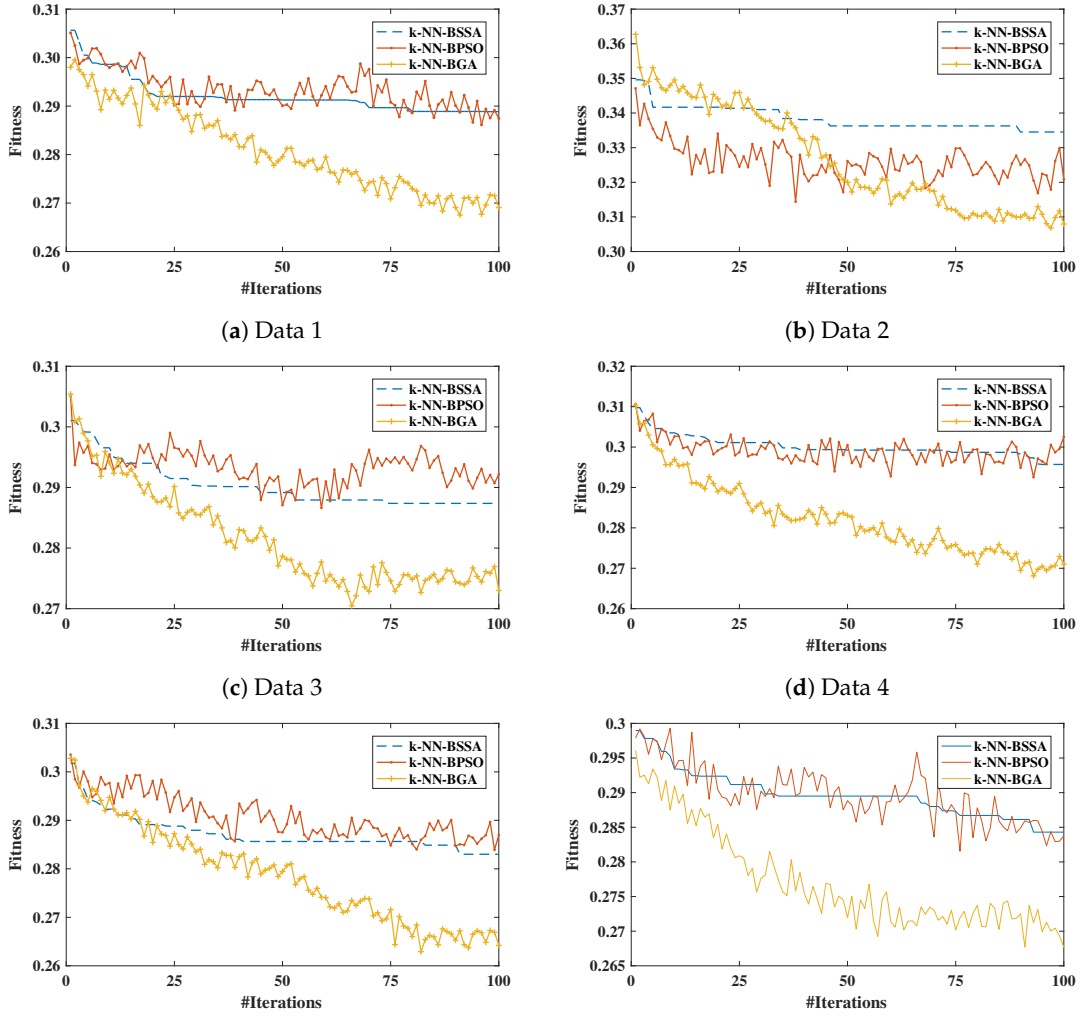
Table 8. Comparison of the classification results for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset with the BoW and Dawson stemming representation (Data 6).

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.7338	0.6111	0.5926	0.6017	400
k-NN-BPSO	0.7393	0.6319	0.5548	0.5908	301.6
k-NN-BGA	**0.7402**	0.6351	0.5510	0.5901	300
k-NN	0.6985	0.5499	0.6144	0.5804	All

The overall results showed the superiority of our model compared to other methods, where k-NN-BGA placed first five times in terms of accuracy, while the standard k-NN obtained first place once. Furthermore, as seen in previous experiments, the best result of all datasets was obtained for Data 5. Therefore, the dataset with the TF-IDF and Lovins

stemming representation techniques (Data 5) was the optimal dataset for our model. Moreover, to ensure the performance of the proposed model (k-NN-BGA) on this dataset, an extra examination and analysis are performed in the next subsection.

The convergence curves of all algorithms and datasets can be found in Figure 4. The figure confirms the better performance of k-NN-BGA against the other algorithms. It is worth mentioning that the convergence curves were calculated according to Equation (7).



(**a**) Data 1 (**b**) Data 2

(**c**) Data 3 (**d**) Data 4

(**e**) Data 5 (**f**) Data 6

Figure 4. Convergence curves for k-NN-BGA and the other methods on the Data 1, Data 2, Data 3, Data 4, Data 5, and Data 6 datasets.

5.2. Comparisons with Other Classification Algorithms

Due it obtaining the best result of all the datasets, an extra experiment was performed on Data 5 to ensure the quality of both the data and classification models. Consequently, in this subsection, three additional classification algorithms were applied to Data 5, namely J48, RF, and SVM. These algorithms were used in this experiment because they are the most popular algorithms in the literature.

Table 9 illustrates the results of the seven classification models, which were k-NN-BSSA, k-NN-BPSO, k-NN-BGA, k-NN, J48, RF, and SVM. In terms of accuracy, k-NN-BGA outperformed the other methods with 0.75%, followed by k-NN-BPSO, k-NN-BSSA, J48, SVM, k-NN, and RF, respectively. k-NN-BGA also exceeded the other methods in terms of precision and the F-measure with 0.66% and 0.60%, respectively. As for the recall measure, k-NN-BSSA obtained the best result, followed by the standard k-NN.

This showed the importance of the feature selection process in such a problem. With only 605 selected features, k-NN-BGA achieved the best results compared to the other methods.

Table 9. Comparison of the best classification results (with the TF-IDF and Lovins stemming representation) for k-NN-BSSA, k-NN-BPSO, and k-NN-BGA with feature selection and the standard k-NN (without feature selection) on the Koirala dataset and other traditional classifiers.

Method	Accuracy	Precision	Recall	F-Measure	No. of Selected Features
k-NN-BSSA	0.7261	0.5962	**0.5974**	0.5968	789.4
k-NN-BPSO	0.7312	0.6198	0.5378	0.5759	616.2
k-NN-BGA	**0.7543**	**0.6622**	0.5633	**0.6088**	**605.4**
k-NN	0.7065	0.5642	0.5936	0.5785	All
J48	0.7229	0.5960	0.5690	0.5822	All
RF	0.7041	0.6342	0.3015	0.4087	All
SVM	0.7075	0.5665	0.5879	0.5770	All

Moreover, compared to the deep learning approach introduced in the literature by Koirala [47], which was conducted on the same dataset, the proposed model presented an optimistic predictability performance with the best accuracy (75.43%), and it reduced the number of features to 303 symmetrical features. On the other hand, the logistic regression approach used in that study reported an accuracy of 75.07% with 797 features. Our proposed approach showed a better performance with a reduced number of symmetrical features.

6. Conclusions

The COVID-19 pandemic is the most critical health crisis, which has negatively affected the lives of people all over the world. During this massively dangerous crisis, a vast amount of misinformation and fake news related to COVID-19 has been rapidly disseminated by different news websites and social media platforms. Such misinformation has caused inaccurate facts to spread among people and increased misconceptions about the pandemic. To detect such COVID-19 misinformation, our study proposed an effective evolutionary fake news detection method, employing four evolutionary models (k-NN-BSSA, k-NN-BPSO, k-NN-BGA with feature selection, and the standard k-NN). The proposed approach aimed to decrease the number of symmetrical features and obtain a high accuracy by implementing three wrapper feature selections (particle swarm optimization (PSO), genetic (GA), and salp swarm algorithm (SSA)). Furthermore, the experiments were implemented on one of the popular datasets: the Koirala dataset. Then, six datasets were constructed using different text tokenization and stemming techniques. Based on the prediction results obtained, the proposed model presented a superior predictability performance with the best accuracy (75.43%) and reduced the number of symmetrical features to 303 features. Compared to other traditional machine learning classifiers, our results showed that the k-NN-BGA model surpassed the accuracy of other classifiers, J48, random forest, and support vector machine respectively.

For future work, the proposed fake detection methodology will be applied to other datasets in other domains. For example, the efficacy of the proposed methodology could be employed to detect fake news in the business sector, the education sector, and many other domains. Furthermore, the performance of the methodology can be investigated on larger datasets.

Author Contributions: B.A.-A. (Conceptualization, Methodology, Related Work, Writing—review & editing, Formal Analysis); A.M.A.-Z. (Conceptualization, Writing—review & editing, Formal Analysis, Visualization); R.A.K. (Writing—review & editing, Methodology, Software, Writing—review & editing, Visualization); I.A. (Supervision; Methodology, Writing—review & editing; Software, Visualization). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data used in this article must be approved by the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

PSO	Particle swarm optimization
GA	Genetic algorithm
SSA	Salp swarm algorithm
DE	Differential evolution
BBO	Biogeography-based optimization
MFO	Moth–flame optimization
BoW	Bag-of-words
TF	Term frequency
TF-IDF	Term frequency-inverse document frequency
kNN	k-nearest neighbors
BSSA	Binary salp swarm algorithm
BPSO	Binary particle swarm optimization
BGA	Binary-coded genetic algorithm
FNN	Forward neural network
LSTM	Long short-term memory

References

1. Sharieh, A.; Khurmah, R.A.; Masadeh, R.; Alzaqebah, A.; Alsharman, N.; Sharieh, F. Effect of Threat Control Management Strategies on Number Infected by COVID-19. In *The Effect of Coronavirus Disease (COVID-19) on Business Intelligence*; Studies in Systems, Decision and Control; Springer: Cham, Switzerland, 2021; Volume 334. [CrossRef]
2. Cheema, G.S.; Hakimov, S.; Ewerth, R. TIB's Visual Analytics Group at MediaEval'20: Detecting Fake News on Corona Virus and 5G Conspiracy. *arXiv* **2021**, arXiv:2101.03529.
3. Elhadad, M.K.; Li, K.F.; Gebali, F. Detecting Misleading Information on COVID-19. *IEEE Access* **2020**, *8*, 165201–165215. [CrossRef]
4. Patwa, P.; Sharma, S.; PYKL, S.; Guptha, V.; Kumari, G.; Akhtar, M.S.; Ekbal, A.; Das, A.; Chakraborty, T. Fighting an infodemic: Covid-19 fake news dataset. *arXiv* **2020**, arXiv:2011.03327.
5. Wani, A.; Joshi, I.; Khandve, S.; Wagh, V.; Joshi, R. Evaluating Deep Learning Approaches for Covid19 Fake News Detection. *arXiv* **2021**, arXiv:2101.04012.
6. Gundapu, S.; Mamid, R. Transformer based Automatic COVID-19 Fake News Detection System. *arXiv* **2021**, arXiv:2101.00180.
7. Aldwairi, M.; Alwahedi, A. Detecting fake news in social media networks. *Procedia Comput. Sci.* **2018**, *141*, 215–222. [CrossRef]
8. Zhuk, D.; Tretiakov, A.; Gordeichuk, A.; Puchkovskaia, A. Methods to identify fake news in social media using artificial intelligence technologies. In Proceedings of the International Conference on Digital Transformation and Global Society, St. Petersburg, Russia, 30 May–1 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 446–454.
9. Atodiresei, C.S.; Tănăselea, A.; Iftene, A. Identifying fake news and fake users on Twitter. *Procedia Comput. Sci.* **2018**, *126*, 451–461. [CrossRef]
10. Deepak, S.; Chitturi, B. Deep neural approach to Fake-News identification. *Procedia Comput. Sci.* **2020**, *167*, 2236–2243.
11. Nyow, N.X.; Chua, H.N. Detecting fake news with tweets' properties. In Proceedings of the 2019 IEEE Conference on Application, Information and Network Security (AINS), Penang, Malaysia, 19–21 November 2019; pp. 24–29.
12. Balaanand, M.; Karthikeyan, N.; Karthik, S.; Varatharajan, R.; Manogaran, G.; Sivaparthipan, C. An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter. *J. Supercomput.* **2019**, *75*, 6085–6105. [CrossRef]
13. Zhang, C.; Gupta, A.; Kauten, C.; Deokar, A.V.; Qin, X. Detecting fake news for reducing misinformation risks using analytics approaches. *Eur. J. Oper. Res.* **2019**, *279*, 1036–1052. [CrossRef]
14. Stahl, K. Fake news detection in social media. *Calif. State Univ. Stanislaus* **2018**, *6*, 4–15.
15. Ozbay, F.A.; Alatas, B. Fake news detection within online social media using supervised artificial intelligence algorithms. *Phys. A Stat. Mech. Appl.* **2020**, *540*, 123174. [CrossRef]
16. Agarwal, A.; Mittal, M.; Pathak, A.; Goyal, L.M. Fake news detection using a blend of neural networks: An application of deep learning. *SN Comput. Sci.* **2020**, *1*, 1–9. [CrossRef]
17. Hosni, A.I.E.; Li, K. Minimizing the influence of rumors during breaking news events in online social networks. *Knowl. Based Syst.* **2020**, *193*, 105452. [CrossRef]

18. Pulido, C.M.; Ruiz-Eugenio, L.; Redondo-Sama, G.; Villarejo-Carballido, B. A new application of social impact in social media for overcoming fake news in health. *Int. J. Environ. Res. Public Health* **2020**, *17*, 2430. [CrossRef] [PubMed]
19. Zhou, X.; Jain, A.; Phoha, V.V.; Zafarani, R. Fake news early detection: A theory-driven model. *Digit. Threat. Res. Pract.* **2020**, *1*, 1–25. [CrossRef]
20. Gangireddy, S.C.R.; Long, C.; Chakraborty, T. Unsupervised Fake News Detection: A Graph-based Approach. In Proceedings of the 31st ACM Conference on Hypertext and Social Media, Virtual Event, New York, NY, USA, 13–15 July 2020; pp. 75–83.
21. Sharif, O.; Hossain, E.; Hoque, M.M. Combating Hostility: Covid-19 Fake News and Hostile Post Detection in Social Media. *arXiv* **2021**, arXiv:2101.03291.
22. Sharma, K.; Seo, S.; Meng, C.; Rambhatla, S.; Liu, Y. Covid-19 on social media: Analyzing misinformation in twitter conversations. *arXiv* **2020**, arXiv:2003.12309.
23. Khurma, R.A.; Aljarah, I.; Sharieh, A.; Mirjalili, S. Evolopy-fs: An open-source nature-inspired optimization framework in python for feature selection. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 131–173.
24. Sadiq, A.S.; Faris, H.; Ala'M, A.Z.; Mirjalili, S.; Ghafoor, K.Z. Fraud detection model based on multi-verse features extraction approach for smart city applications. In *Smart Cities Cybersecurity and Privacy*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 241–251.
25. Ala'M, A.Z.; Hassonah, M.A.; Heidari, A.A.; Faris, H.; Mafarja, M.; Aljarah, I. Evolutionary competitive swarm exploring optimal support vector machines and feature weighting. *Soft Comput.* **2021**, *25*, 3335–3352.
26. Khurma, R.A.; Aljarah, I.; Sharieh, A. A Simultaneous Moth Flame Optimizer Feature Selection Approach Based on Levy Flight and Selection Operators for Medical Diagnosis. *Arab. J. Sci. Eng.* **2021**, 1–26.
27. Khurmaa, R.A.; Aljarah, I.; Sharieh, A. An intelligent feature selection approach based on moth flame optimization for medical diagnosis. *Neural Comput. Appl.* **2020**, 33, 7165–7204. [CrossRef]
28. Khurma, R.A.; Castillo, P.A.; Sharieh, A.; Aljarah, I. Feature Selection using Binary Moth Flame Optimization with Time Varying Flames Strategies. In Proceedings of the 12th International Joint Conference on Computational Intelligence-Volume 1: ECTA, INSTICC, Lisbon, Portugal, 2–4 November 2020; pp. 17–27. [CrossRef]
29. Khurma, R.A.; Sabri, K.E.; Castillo, P.A.; Aljarah, I. Salp Swarm Optimization Search Based Feature Selection for Enhanced Phishing Websites Detection. In Proceedings of the Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, Seville, Spain, 7–9 April 2021; Springer Nature: Berlin/Heidelberg, Germany, 2021; Volume 12694, p. 146.
30. Khurma, R.A.; Aljarah, I.; Sharieh, A. Rank based moth flame optimisation for feature selection in the medical application. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
31. Khurma, R.A.; Aljarah, I.; Sharieh, A. An Efficient Moth Flame Optimization Algorithm using Chaotic Maps for Feature Selection in the Medical Applications. In Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods-Volume 1: ICPRAM, INSTICC, Valletta, Malta, 22–24 February 2020; pp. 175–182. [CrossRef]
32. Pernkopf, F.; O'Leary, P. Feature selection for classification using genetic algorithms with a novel encoding. In Proceedings of the International Conference on Computer Analysis of Images and Patterns, Warsaw, Poland, 5–7 September 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 161–168.
33. Khurma, R.A.; Castillo, P.A.; Sharieh, A.; Aljarah, I. New Fitness Functions in Binary Harris Hawks Optimization for Gene Selection in Microarray Datasets. In Proceedings of the 12th International Joint Conference on Computational Intelligence-Volume 1: ECTA, INSTICC, Lisbon, Portugal, 12–14 November 2020; pp. 139–146. [CrossRef]
34. Khurma, R.A.; Aljarah, I. A Review of Multiobjective Evolutionary Algorithms for Data Clustering Problems. *Evol. Data Clust. Algorithms Appl.* **2021**, 177. [CrossRef]
35. Qian, S.; Singer, Y. Fast parallel algorithms for feature selection. *arXiv* **2019**, arXiv:1903.02656.
36. Faris, H.; Alqatawna, J.; Ala'M, A.Z.; Aljarah, I. Improving email spam detection using content based feature engineering approach. In Proceedings of the 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, Jordan, 11–13 October 2017; pp. 1–6.
37. Al-Zoubi, A.; Alqatawna, J.; Faris, H.; Hassonah, M.A. Spam profiles detection on social networks using computational intelligence methods: The effect of the lingual context. *J. Inf. Sci.* **2019**, *47*, 58–81. [CrossRef]
38. Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 43–55.
39. Al-Qudah, D.A.; Ala'M, A.Z.; Castillo-Valdivieso, P.A.; Faris, H. Sentiment Analysis for e-Payment Service Providers Using Evolutionary eXtreme Gradient Boosting. *IEEE Access* **2020**, *8*, 189930–189944. [CrossRef]
40. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* **2021**, *25*, 5277–5298. [CrossRef]
41. Zhang, Y.; Gu, X. Biogeography-based optimization algorithm for large-scale multistage batch plant scheduling. *Expert Syst. Appl.* **2020**, *162*, 113776. [CrossRef]
42. Wang, F.; Zhang, H.; Zhou, A. A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm Evol. Comput.* **2021**, *60*, 100808. [CrossRef]

43. Ala'M, A.Z.; Heidari, A.A.; Habib, M.; Faris, H.; Aljarah, I.; Hassonah, M.A. Salp chain-based optimization of support vector machines and feature weighting for medical diagnostic information systems. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 11–34.
44. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [CrossRef]
45. 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.
46. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSA: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [CrossRef]
47. Koirala, A. COVID-19 Fake News Classification with Deep Learning. Master's Thesis, Asian Institute of Technology, Bangkok, Thailand, 2020.
48. Hung, L.P.; Alfred, R. A performance comparison of feature extraction methods for sentiment analysis. In *Asian Conference on Intelligent Information and Database Systems*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 379–390.
49. Habernal, I.; Ptáček, T.; Steinberger, J. Supervised sentiment analysis in Czech social media. *Inf. Process. Manag.* **2014**, *50*, 693–707. [CrossRef]
50. Ghosh, M.; Sanyal, G. Preprocessing and feature selection approach for efficient sentiment analysis on product reviews. In *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Singapore*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 721–730.
51. Singh, J.; Gupta, V. A novel unsupervised corpus-based stemming technique using lexicon and corpus statistics. *Knowl. Based Syst.* **2019**, *180*, 147–162. [CrossRef]
52. Abdolahi, M.; Zahedh, M. Sentence matrix normalization using most likely n-grams vector. In Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22–22 December 2017; pp. 40–45.
53. Hassonah, M.A.; Al-Sayyed, R.; Rodan, A.; Ala'M, A.Z.; Aljarah, I.; Faris, H. An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowl. Based Syst.* **2020**, *192*, 105353. [CrossRef]
54. 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**, 0165551520917651. [CrossRef]