algorithms

MDPI

Article

Detecting Fake Accounts on Instagram Using Machine Learning and Hybrid Optimization Algorithms

Pegah Azami and Kalpdrum Passi *

School of Engineering and Computer Science, Laurentian University, Sudbury, ON P3E 2C6, Canada; pazami@laurentian.ca
* Correspondence: kpassi@laurentian.ca

Abstract: In this paper, we propose a hybrid method for detecting fake accounts on Instagram by using the Binary Grey Wolf Optimization (BGWO) and Particle Swarm Optimization (PSO) algorithms. By combining these two algorithms, we aim to leverage their complementary strengths and enhance the overall optimization performance. We evaluate the proposed hybrid method using four classifiers: Artificial Neural Network (ANN), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression (LR). The dataset for the experiments contains 65,329 Instagram accounts. We extract features from each account, including profile information, posting behavior, and engagement metrics. The Binary Grey Wolf and Particle Swarm Optimizations, when combined to form a hybrid method (BGWOPSO), improved the performance in accurately detecting fake accounts on Instagram.

Keywords: fake accounts; Binary Grey Wolf Optimization; Particle Swarm Optimization; Support Vector Machine; Logistic Regression; K-Nearest Neighbor; Artificial Neural Network

check for updates

Citation: Azami, P.; Passi, K. Detecting Fake Accounts on Instagram Using Machine Learning and Hybrid Optimization Algorithms. *Algorithms* **2024**, *17*, 425. https://doi.org/10.3390/a17100425

Academic Editors: Grigorios Beligiannis, Efstratios F. Georgopoulos, Spiridon D. Likothanassis, Isidoros Perikos and Ioannis X. Tassopoulos

Received: 9 July 2024
Revised: 19 September 2024
Accepted: 21 September 2024
Published: 24 September 2024

Copyright: © 2024 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

Social media has made huge inroads in the lives of people by increasing the reach for sharing information across the globe, which has prompted several investments in the sector to generate higher profits. In this way, human social life is also tied to online social networks. Social life has undergone great changes with the emergence of these sites. Fake user accounts in social media introduce themselves as real people with the aim of destroying customer loyalty and creating mistrust among users.

Online social media, such as Facebook, Twitter, or LinkedIn, allow users to introduce themselves through an online profile. With these profiles, users can create a wide range of online social relationships. Due to the open nature of online social networks, users can appear in the network with multiple identities. Therefore, verifying the identity of users is considered one of the important issues for security and privacy. In order to set up social relations in a secure way, the identity of users must be authenticated to prevent the creation of unreal (fake) communications on a large scale. Users can create multiple profiles on various social media platforms, Instagram in particular, which makes it difficult to identify the correct profile of the users and thereby is incompetent in user authentication. The consequence of such options allows users to create fake profiles to enter into the profiles of unsolicited users and cause harm to unsuspicious users [1].

The number of fake accounts on social media such as Instagram has increased manifold due to the popularity of these platforms in the last decade. Since fake accounts create an identity obfuscation, extortionists are able to mislead other users into losing their reputation or personal valuables, thereby motivating us to identify fake accounts accurately. Fake account holders also aim to capitalize on users' sympathies by spreading fake news and receiving money from innocent people. Multiple fake accounts may be created that do not belong to anyone and are used only to vote in online voting systems or earn referral points in online games.

Researchers have been at the forefront in proposing different techniques to detect fake accounts on social networks, thereby increasing interest in this area of research. These techniques include the use of feature selection and machine learning algorithms. Fake account creators are able to circumvent the defensive techniques on platforms to mislead unsuspecting users. There is a need to constantly monitor and devise stronger techniques to detect fake accounts as the attackers become more skilled at deception. The main concerns in the diagnosis process are the accuracy and response time of feature analysis.

Accounts are generally identified by the user's picture, a username, and a brief biography of the user on Instagram. Instagram does not provide any method to validate the real identity of a user based on their profile. The lack of protective tools on these social media accounts has provided the ground for malicious actions by malicious people. Fake accounts on social media are generally identified based on the user profiles and by detecting unusual activity such as a sudden surge in the number of followers [2]. At first glance, it is difficult to prevent such scenarios, however, machine learning provides the techniques and the tools to detect fake accounts.

Before machine learning can be applied to datasets, it is almost always necessary to preprocess the data into a form that can provide optimal results, and the most successful method is to select the right features in the dataset that are effective in predicting the target variable. Not all features are important in predicting the target variable, and some features may increase the computation speed in the prediction process and as a result, might be detrimental in accurately predicting the outcome. The raw data must be filtered by eliminating the features that are not helpful in predicting the target variable through selection of the right features [3,4]. Selecting the right features is not only important but requires different techniques, some of which can be expensive computationally. One of the techniques to select the features is through different metaheuristics to find an optimal solution, here used for finding the right set of features. Some of the metaheuristic algorithms include Grey Wolf Optimization (GWO) [5,6], genetic algorithms (GAs), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) [7,8], and Differential Evolution (DE) [9]. We have applied and tested GWO and PSO algorithms to find the best features to be input into a machine learning model to obtain the most accurate prediction of fake accounts.

The PSO algorithm is inspired by the behavior of particles in a system where they tend to accumulate in concentrated locations [10]. The particles search for the best positioning locally and globally by looking for surrounding particles nearby and then forming the swarm in the whole system to find the best position globally. Particles float around in a multidimensional search space [11]. Using knowledge of the neighbors, the particles change their positions to find the best location. GWO is also a nature-inspired technique that has been used widely in a number of optimization problems. The GWO algorithm is based on the hunting of prey by grey wolf packs—which consist of different types of members in a hierarchy—and how the members are assigned different tasks and importance in the hunting process [2]. In this study, we have introduced a hybrid algorithm, BGWOPSO, by combining the Binary Grey Wolf Optimization algorithm (BGWO) with Particle Swarm Optimization (PSO) to find the best feature in the Instagram dataset. This hybrid method (BGWOPSO) is also considered a multi-objective function, the goals of this algorithm are a trade-off between minimizing the error and increasing the accuracy and recall in the fake account detection system. Classification techniques are applied to measure and compare the optimization techniques, which utilize many parameters to describe each object. Four classification methods that are tested include the Support Vector Machine (SVM), K-Nearest Neighbor, Neural Networks (NN), and Logistic Regression. The classifiers are evaluated by the measure of accuracy, precision, recall, and AUC (area under ROC) values.

Considering the importance of the problem in detecting fake accounts on social networks and after analyzing the literature, the following solution is presented to answer this problem:

- Identify fake accounts with high precision;
- Reduce false alarms while detecting fake accounts;

- Propose a hybrid optimization technique to improve the accuracy of fake account detection.

2. Related Work

Online social networks such as Instagram, Facebook, Google Plus, Twitter (X), LinkedIn, Pinterest, etc., are growing platforms [12] because users can utilize these platforms, without paying anything, to communicate with other users and share their information [13]. The current statistics show that there are about 349 million Facebook users in India and 194 million registered users in America [14]. The same trend has been found for Twitter, which has about 145 million daily users and 330 million monthly users. A similar type of trend has also been observed on Instagram and YouTube. Discovering and removing fake users in social networks improves the security of users in these networks and increases interest in using these networks. Therefore, this issue has become one of the most challenging research issues in the field of social networks, and many researchers have presented methods to solve this problem.

A number of researchers have applied machine learning techniques to various Online Social Network (OSN) platforms such as Twitter (X), Facebook, and Instagram. For example, SVM, Random Forest, Logistic Regression, J48, Naïve Bayes, Neural Network, and KNN methods have been used to detect fake Twitter accounts [2,15–26]. Equally, feature reduction and correlation techniques have been used to increase the accuracy of the classifiers in [2,15,23,25,26].

Similarly, for the detection of fake accounts on Facebook, several classifiers have been used by researchers, including SVM, Random Forest, Neural Networks, Naïve Bayes, and KNN, as seen in [27–30]. Entropy and information gain was applied in [29] to reduce the features.

Fake accounts on Instagram have been identified using various machine learning methods, including Random Forest, Logistic Regression, Naïve Bayes, SVM, Neural Networks, and KNN, as seen in [3,4,6–8,31,32]. The Instagram dataset with fake accounts was purchased from different sources and authentic users in [3]. Based on these sources, seventeen features have been previously used, as follows: six metadata features, three media information features, two interaction features, two media tag features, and four media similarity features. In addition, a cost-sensitive genetic algorithm to deal with abnormal orientation in the dataset was used in [31]; equally, an approach combining image recognition using CNN and natural language processing was used to identify fake accounts on Instagram in [6]. A dataset of legitimate and fake accounts was first created in [8], and classifiers were then applied to detect fake accounts. Multiple stages of Instagram data processing were applied in [32], which include data preprocessing, model selection, and evaluation of the classification algorithms.

Table 1 gives a comprehensive explanation of the datasets, performance, and limitations of these works.

Table 1. Summary of the literature.

Reference	Data	Methods and Performance	Limitations and Research Gaps
Khaled et al. [15]	Twitter (X) accounts and bots	Feature selection with SVM, Neural Networks, and a combination of SVM-NN with accuracy of 98% in training dataset	Efficiency in the test dataset is lower than training dataset
Kondeti et al. [16]	Twitter (X) accounts and bots	SVM, Logistic Regression, Random Forest, and KNN with a peak accuracy of 98%. Used Z-Score and Min–Max normalization	Small dataset and very few features
Suganya et al. [17]	Twitter (X) accounts	SVM, Decision Tree, Random Forest, Naïve Bayes with an accuracy of 97% for SVM	Small dataset and very few features

Table 1. *Cont.*

Reference	Data	Methods and Performance	Limitations and Research Gaps
Kaubiyal and Jain [2]	Twitter (X) accounts	Logistic Regression, Random Forest, and SVM with a maximum accuracy of 97.9% by Random Forest	Imbalanced data with large number of genuine accounts and spambots
Bindu, Mishra, and Thilagam [18]	Twitter (X) accounts	Spammer communities on Twitter using unsupervised method	
Patil et al. [19]	Twitter (X) accounts	Propose a model for detection	Implementation and results are not given
Prabhu Kavin et al. [20]	Twitter (X) accounts	Spam detection using Logistic Regression, SVM, Random Forest, Neural Networks. Images were used to identify inappropriate content and fake images	Did not detect fake accounts
Kadhim and Abdullah [21]	Twitter (X) accounts Management Information Base "MIB" dataset	Feature selection with Spearman's correlation coefficient and chi-square test. Random Forest, SVM, and Naïve Bayes algorithms as a stack ensemble method were applied and Logistic Regression was used as meta classifier with a combined accuracy of 97.8%	A bigger dataset would give a better understanding of the methods applied
Benabbou, Boukhouima, and Sael [22]	Twitter user profiles as legitimate or fake	Bidirectional Gated Recurrent Unit (BiGRU) model proposed and compared with LSTM and CNN with an accuracy of 99.44%	Tweets were gathered in a single file and transformed into a vector space using the GloVe word-embedding technique in order to preserve the semantic and syntactic contexts
David, Siordia, and Moctezuma [23]	Twitter (X) accounts	71 descriptive features were extracted from the profiles. Random Forest and Naïve Bayes gave an accuracy of 94% and 91%, respectively	Less than 432 followers and 1.8% of responses were secured
P, Sowmya and Chatterjee, Madhumita [24]	Data extraction from Facebook and Twitter	Similarity between the attributes and the network was used to detect cloned and fake profiles	Results are not presented
Bharti and S. Pandey [25]	Twitter (X) accounts	Logistic Regression and Particle Swarm Optimization to classify accounts to real and fake with an accuracy of 96%	Requires more features to be tested
Homsi et al. [26]	Twitter (X) accounts, data from My Information Bubble (MIB) [33]	Random Forest, J48, Naïve Bayes, and KNN were used as machine learning algorithms. On the other hand, PCA and correlation were two reduction techniques. An accuracy of 98.6% was achieved with Random Forest and correlation method	More features to be added
SudalaiMuthu et al. [27]	Facebook dataset	Random Forest, Neural Networks, and SVM with an accuracy of 87%	Limited features
Awan et al. [28]	4000 Facebook accounts	Random Forest with 93% accuracy	Limited features
Munga and Mohandas [29]	Facebook accounts with limited profile information	Feature selection using entropy and information gain. Decision Tree, Naïve Bayes, and Random Forest were used without feature selection. Neural Networks, Random Forest, SVM were used with information gain. Highest accuracy was achieved with SVM and information gain with 99.64% accuracy	Limited features

Table 1. *Cont.*

Reference	Data	Methods and Performance	Limitations and Research Gaps
Gupta and Kaushal [30]	OSN Facebook accounts	12 machine learning classifiers were used for accuracy testing, including SVM, minimum-order optimization, Naïve Bayes, KNN, Decision Tree, Random Forest, with highest accuracy of 79%	Limited access to data due to privacy and security
Purba, Asirvatham, and Murugesan [3]	Instagram dataset included fake accounts purchased from different sources and authentic users	Random Forest algorithm produced the highest accuracy for the classification of 2 classes (authentic, fake) and 4 classes (authentic, active fake user, inactive fake user, spam) with an accuracy of 91.76%	Performance improvement desired
Dey et al. [4]	Instagram dataset	Logistic Regression and Random Forest accuracy of 90.8% and 92.5%, respectively	Performance improvement desired
Akyon and Esat Kalfaoglu [31]	Instagram dataset	Naïve Bayes, Logistic Regression, Support Vector Machine, and Neural Network, with genetic algorithm giving the highest accuracy of 96%	Biased features in the automated account dataset
Saranya Shree, Subhiksha, and Subhashini [6]	Instagram web-scraping dataset, labeled for training	Combining image recognition and natural language processing to identify fake accounts. CNN gave an accuracy of 91.5%	Limited accounts and performance
Meshram et al. [7]	Collected 1002 real Instagram profiles and 201 fake profiles through web crawler	Neural Networks, Random Forest, and Logistic Regression. Random Forest gives highest accuracy of 96.94%	Detection of the fake profiles and automatic money owed
Sheikhi [8]	A dataset of legitimate and fake accounts was extracted on Instagram using a crawler	Bagged Decision Tree was the proposed method that was compared to Random Tree, J48, SVM, Naïve Bayes, and Hoeffding Tree. Bagged Tree achieved an accuracy of 98.45%	Data and features were extracted by the crawler to create a dataset
Durga and Sudhakar [32]	Instagram dataset	KNN, Logistic Regression, and Decision Tree were used and the best accuracy of 96% was achieved by Decision Tree.	Small dataset and few features

3. Materials and Methods

The dataset was taken from Kaggle and contains 65,329 rows, consisting of 32,869 fake accounts and 32,460 real accounts. It contains 17 features and a label column (fake/real), labeled as 0 and 1. We describe the methodology in Section 3.1 briefly.

3.1. Methodology

The methodology used in the present research for fake account detection in an Instagram dataset is described in the flowchart in Figure 1. From the literature review described in Table 1, it was observed that the feature selection applied in [15,16,21,25,26,29] improved performance when applied as a preprocessing step before using the machine learning methods. In these studies, genetic algorithms and Particle Swarm Optimization (PSO) algorithms were used to select optimal features to achieve high accuracy in prediction. Motivated by the results from these studies, we applied Particle Swarm Optimization (PSO) and Grey Wolf Optimization to select the optimal features in the Instagram dataset to improve the accuracy of predicting fake accounts. Further, we propose a hybrid model to fine-tune the parameters of PSO with Binary Grey Wolf Optimization (BGWO), which further improves accuracy in detecting fake Instagram accounts. The dataset obtained after

applying the optimization methods was subjected to four different classifiers, namely SVM, KNN, Neural Networks, and Logistic Regression. Four models were used for training and testing ratios, namely 90:10, 80:20, 70:30, 60:40 to train and test the classifiers. The models were implemented using MATLAB 7.0 on an Intel Core i7 processor with 16 GB RAM. The performance of the models was measured using accuracy, precision, recall, and area under the ROC curve (AUC).

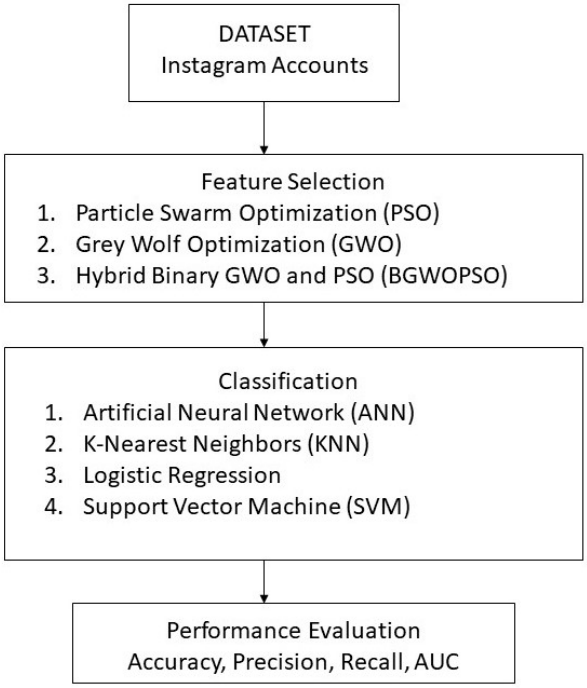

Figure 1. Flowchart of the methodology.

3.2. Feature Selection Using Optimization Methods

In this research, we used two optimization methods—namely Particle Swarm Optimization (PSO) [9] and Grey Wolf Optimization [10]—on the Instagram data. PSO is a metaheuristic optimization algorithm that can be used for feature selection. The algorithm optimizes a fitness function by iteratively adjusting a set of candidate solutions (in this case, the selected features). PSO selects the best features based on a fitness function. Mirjalili et al. [34] proposed the Grey Wolf Optimization (GWO) algorithm, which is inspired by the technique used by grey wolves in finding their prey optimally. The wolves are classified into four types by leadership hierarchy, namely, alpha, beta, gamma, and delta. The hunting behavior includes searching for prey, encircling prey, and attacking prey. Since feature selection is a binary concept, Binary Grey Wolf Optimization (BGWO) is used for hybridization.

3.2.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart [35] in 1995; it is a population-based stochastic optimization algorithm based on swarm behaviors found in nature such as in schools of fish and flocks of birds. In PSO, a population of candidate solutions, called particles, moves in search of the optimal solution to a given problem. Feature selection using PSO involves selecting a subset of features that can best represent the data for a particular task, such as classification or regression. The objective function used in PSO for feature selection is often based on a performance metric of the classification or regression model, such as accuracy or mean squared error.

The PSO algorithm starts by initializing a population of particles, each of which represents a candidate solution to the feature selection problem. Each particle is associated with a velocity and a position in the search space. The velocity and position of each particle

are updated based on its own best-known solution and the best-known solution of its neighbors in the search space. During each iteration of the PSO algorithm, the particles move through the search space, and the fitness of each particle is evaluated based on the performance metric of the classification or regression model. The fitness of each particle is then used to update its velocity and position in order to converge toward the optimal solution. A maximum number of iterations or a criterion of convergence is used to control the termination point. The PSO algorithm works as follows:

Initialization: particles identified by their position and velocity are initialized;

Evaluation: each particle is evaluated by its fitness function;

Update of the particle's velocity and position: Each particle updates its velocity and position based on its own best-known position (i.e., the position that resulted in the best fitness so far) and the best-known position of the swarm. The new velocity and position are computed using Equations (1) and (2):

$$v_i^{t+1} = v_i^t + c_1 r_1 \left(pbest_i^t - x_i^t \right) + c_2 r_2 \left(g_{best} - x_i^t \right) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

where v_i is the velocity; c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are random numbers between 0 and 1; $pbest_i^t$ is the best-known position of the particle; g_{best} is the best-known position of the swarm; and x_i^t is the current position of the particle;

Termination: the algorithm continues to iterate until a termination criterion is met, such as a maximum number of iterations or the convergence of the solution.

3.2.2. Grey Wolf Optimization (GWO) Algorithm

Mirjalili et al. [34] proposed the Grey Wolf Optimization (GWO) algorithm, which is inspired by the method adopted by grey wolves in searching, encircling, and attacking prey through a hierarchical formulation of their members. The hierarchy consists of alpha, beta, delta, and omega wolves. The top of the hierarchical order is the alpha wolf that makes the decisions and instructs the other wolves. The next in the hierarchy is the beta wolf, which follows the instructions of the alpha wolf and gives instructions to the other wolves. The third in the hierarchy is the delta wolves. The lowest in the hierarchy is the omega wolf that follows the instructions of all the wolves above in the hierarchy. Each wolf in the hierarchy plays a different role by looking for danger and alerting the others or guarding the other wolves from potential danger. A mathematical model is used to optimize the hunting behavior of grey wolves.

Since alpha (α) is the top wolf in the hierarchy, we model the most appropriate solution as alpha (α). As a result, among the best solutions, we name the second and third ones as beta (β) and delta (δ). The rest of the candidate solutions of omega (ω) are considered. In the grey wolf optimizer algorithm, α, β, and δ guide the hunting process (optimization) whereas ω wolves follow these three groups. The equations representing the encircling behavior are given by (3) and (4).

$$x^{t+1} = x_p^t - b.d \quad (3)$$

$$d = \left| e.x_p^t - x^t \right| \quad (4)$$

In Equations (3) and (4), t represents the number of iterations, b and e represent coefficients, x_p^t is the prey's positions vector, and x^t represents the position vector of a grey wolf. Then, b and e are calculated according to Equations (5) and (6).

$$b = 2 * a * r_1 - a \quad (5)$$

$$e = 2.r_2 \quad (6)$$

In Equations (5) and (6), a decrease from 2 to 0 linearly over iterations—proportional to the number of iterations for each dimension—is given by Equation (7), and r_1 and r_2 are random vectors in the interval [0, 1].

$$a = 2 - t \cdot \frac{2}{\underset{i}{\max} ter} \tag{7}$$

where t is the iteration number, ter is the optimization's total number of iterations.

Alpha (α) represents the optimal solution as it is on the top of the hierarchy, beta (β) and delta (δ) represent the second-best and third-best solutions, respectively. The three best solutions prompt omega (ω) to change its position according to the positions of alpha, beta, and delta. The positions are updated according to Equation (8).

$$x^t = \frac{x_1^t + x_2^t + x_3^t}{3} \tag{8}$$

In Equation (8), $x_1^t,\ x_2^t,\ x_3^t$ are the three best solutions at a given iteration t and are given by Equation (9):

$$x_1^t = x_\alpha^t - b_1.d_\alpha x_2^t = x_\beta^t - b_2.d_\beta x_3^t = x_\delta^t - b_3.d_\delta \tag{9}$$

In Equation (9), $b_1,\ b_2,\ b_3$ are calculated as in Equation (5) and $d_\alpha,\ d_\beta,\ d_\delta$ as given in Equation (4) can be described by Equation (10).

$$d_\alpha = \left|e_1.x_\alpha^t - x^t\right| d_\beta = \left|e_2.x_\beta^t - x^t\right| d_\delta = \left|e_3.x_\delta^t - x^t\right| \tag{10}$$

where $e_1,\ e_2,\ e_3$ are calculated based on Equation (6).

Binary GWO for Feature Selection

Feature selection can be considered a binary problem by either selecting a feature or discarding it and, thus, Binary GWO is described to select the features. According to [35], the updating mechanism of wolves is a function of three positions, namely $x_1^t, x_2^t,\ x_3^t$, which represent the best three solutions. Equation (8) is modified to transform the result into a binary choice, as given by Equation (11).

$$x^t = \begin{cases} 1,\ if\ simoid\left(\frac{x_1^t + x_2^t + x_3^t}{3}\right) \geq rand \\ 0,\ otherwise \end{cases} \tag{11}$$

where x^t results into a value of 1 if the sigmoid function (see Equation (12)) is greater than a random number generated in the interval [0, 1] and 0 otherwise, for iteration t.

$$\text{sigmoid(x)} = \frac{1}{1 + e^{-10(x-0.5)}} \tag{12}$$

In Equation (9), x_1^t, x_2^t, x_3^t are modified, as in Equation (13).

$$x_1^t = \begin{cases} 1\ if\ \left(x_\alpha^t + bsetp_\alpha^t\right) \geq 1 \\ 0\ othwewise \end{cases} x_2^t = \begin{cases} 1\ if\ \left(x_\beta^t + bstep_\beta^t\right) \geq 1 \\ 0\ othwewise \end{cases} x_3^t = \begin{cases} 1\ if\ \left(x_\delta^t + bstep_\delta^t\right) \geq 1 \\ 0\ othwewise \end{cases} \tag{13}$$

where $x_\alpha^t,\ x_\beta^t,\ x_\delta^t$ represent the position vectors for alpha, beta, and delta wolves in iteration t, and $bstep_{\alpha,\beta,\delta}^t$ is a binary step in iteration t, as given by Equation (14).

$$bstep\ _{\alpha,\beta,\delta}^{\ t} = \begin{cases} 1\ if\ \left(cstep_{\alpha,\beta,\delta}^t\right) \geq rand \\ 0\ othwewise \end{cases} \tag{14}$$

where *rand* is a random value in the interval [0, 1] derived from a uniform distribution, t is the iteration, and $cstep^{t}_{\alpha,\beta,\delta}$ is the continuous value in iteration t, given by Equation (15).

$$cstep^{t}_{\alpha,\beta,\delta} = \frac{1}{1 + e^{-10(b.d_{\alpha,\beta,\delta} - 0.5)}} \tag{15}$$

3.2.3. Hybridization of Binary GWO and PSO

Many researchers have presented several hybridizations for heuristic variants. Hybridization can be performed by coevolutionary techniques either at a low level or a high level [36]. We merge the functionalities of BGWO and PSO by coevolutionary technique using low-level hybridization. The hybridization generates two distinct variants of the solution resulting in a mixed hybrid.

In hybridizing BGWO with PSO, the positions of the first three agents are updated in the search space by the proposed mathematical Equation (16). As an alternative to employing the usual mathematical equations, we control the exploration and exploitation of the grey wolf in the search space by inertia constant w.

$$\boldsymbol{d_{\alpha}} = \left|\boldsymbol{e_1}.\boldsymbol{x^t_{\alpha}} - \boldsymbol{w}*\boldsymbol{x^t}\right| d_{\beta} = \left|e_2.x^t_{\beta} - w * x^t\right| d_{\delta} = \left|e_3.x^t_{\delta} - w * x^t\right| \tag{16}$$

In order to combine the BGWO and PSO, the equation of the updated velocity [37] is given in Equation (17), where an inertia constant weight w is used to control the velocity in the (t + 1)th iteration and x^t_1, x^t_2, x^t_3 are as defined in (13).

$$v^{t+1}_i = w * v^t_i + c_1 r_1 \left(x^t_1 - x^t_i\right) + c_2 r_2 \left(x^t_2 - x^t_i\right) + c_3 r_3 \left(x^t_3 - x^k_i\right) \tag{17}$$

$$x^{t+1}_i = x^t_i + v^{t+1}_i \tag{18}$$

where x^t_i and v^{k+1}_i are calculated based on Equation (11) and Equation (17), respectively.

The solution is one-dimensional with a length equal to the number of features having values of 1 (feature selected) and 0 (feature not selected). In order to find the minimum number of features using BGWOPSO and to maximize the classification accuracy, the fitness function [35] is defined in Equation (19).

$$fitness = \alpha \rho_R(D) + \beta \frac{|S|}{|T|} \tag{19}$$

where α = [0, 1] and $\beta = (1 - \alpha)$ are parameters adapted from [35]; $\rho_R(D)$ indicates the error rate of the classifier; $|S|$ is the selected subset of features; and $|T|$ represents all of the features in the dataset.

3.3. Classification Methods

Classification is a supervised learning process where labeled data are trained to predict unknown or unseen data. The class labels are determined for the test data and compared with the label of the tuple to determine the accuracy of the correct labeling. The trained model can be used to classify new datasets. It can also be used for future predictions by determining the view that is compatible with it. In today's world, discussion of the classification of information is of great importance; this ensures that a suitable model can be obtained for the analysis of specific data, and its behavior pattern can also be determined by a preliminary examination of the characteristics of the data. In classification problems, the goal is to identify the characteristics that show the group to which each item belongs. This model can be used both to understand the existing data and to predict the behavior of new cases. In data mining, the subject of information classification examines such models and methods. In information classification, the goal is to obtain a model for understanding the behavior pattern and characteristics of a set of data, so that with its help and without knowing the behavior of an entity—according to its characteristics and using the obtained

model—it can recognize the behavior and classify that entity into a special group. Today, many companies around the world use this science to analyze, investigate, and predict the behavior of their customers. The classifiers used in this research include the Support Vector Machine (SVM), Neural Networks, K-Nearest Neighbors, and Linear Regression.

4. Results

Particle Swarm Optimization (PSO), Binary Grey Wolf Optimization (BGWO), and hybrid BGWOPSO optimization algorithms were applied and tested on four classifiers, namely SVM, KNN, Neural Networks, and Logistic Regression. The classifiers were trained and tested on four different ratios (90:10, 80:20, 70:30, and 60:40).

Table 2 shows the results of the detection of fake accounts in the Instagram dataset using BGWO, PSO, and BGWOPSO optimization algorithms with four different classifiers. Using ANN, a 99.1% accuracy and an AUC of 1.0 was achieved with BGWOPSO optimization for the training–testing ratio 80:20. Using KNN, the highest accuracy of 98.25% was achieved with BGWOPSO and the highest AUC of 0.9971 was achieved with BGWO optimization for the training–testing ratio 90:10. For SVM, the highest accuracy of 97.92% and AUC of 0.9887 were obtained for the training–testing ratio 80:20 with BGWOPSO. For the Logistic Regression, the highest accuracy of 92.55% and AUC of 0.9336 were achieved for the training–testing ratio 80:20 with the BGWOPSO optimization algorithm.

Table 2. Performance of each classifier with optimization techniques on Instagram dataset.

Ratio	**60:40**		**70:30**		**80:20**		**90:10**	
Optimization	**Accuracy**	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**
Classifier: SVM								
PSO	89.75	91.18	89.35	90.42	88.55	89.54	87.68	91.34
BGWO	92.96	93.45	86.64	89.23	89.96	92.34	87.44	89.88
BGWOPSO	93.42	94.48	92.56	95.23	97.92	98.87	96.45	97.25
Classifier: K-Nearest Neighbor								
PSO	85.32	89.65	88.74	91.83	83.42	91.35	82.90	87.35
BGWO	88.24	93.75	90.65	92.34	89.25	94.64	92.95	99.71
BGWOPSO	91.22	94.54	93.96	96.38	96.27	98.28	98.25	94.12
Classifier: Artificial Neural Network								
PSO	88.52	90.11	85.27	91.84	96.24	97.24	85.36	92.88
BGWO	85.23	92.73	94.49	96.15	97.5	98.93	92.64	94.33
BGWOPSO	92.74	93.47	95.89	98.75	99.1	1.0	93.69	96.35
Classifier: Logistic Regression								
PSO	83.24	86.27	83.12	87.46	89.78	90.75	87.54	89.18
BGWO	86.75	88.56	88.43	90.36	90.23	91.78	88.23	90.55
BGWOPSO	89.74	90.57	90.17	93.12	92.55	93.36	90.14	92.23

4.1. Summary of Results

Table 3 presents the best performance of the classifiers with different optimization techniques. Four classification algorithms were implemented with PSO, BGWO, and BGWOPSO and were tested for accuracy and the AUC scores with different training and testing ratios. The results in Table 3 show the best performance of the optimization techniques and the classifiers. It can be observed from Table 2 that BGWOPSO was the best optimizer for all classifiers such as SVM, K-Nearest neighbors, ANN, and Logistic Regression.

Table 3. Summary of best performance of classifiers and optimization techniques.

Classifier	Optimization Technique	Ratio	Accuracy (%)	AUC
SVM	BGWOPSO	80:20	97.92	0.9887
K-Nearest Neighbor	BGWOPSO	90:10	98.25	0.9412
K-Nearest Neighbor	BGWO	90:10	92.95	0.9971
ANN	BGWOPSO	80:20	99.1	1
Logistic Regression	BGWOPSO	80:20	92.55	93.36

The ROC curve is the representation of the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). The accuracy of the classifiers can be determined by the curves, those that are closer to the upper left corner perform better. In general, the classifiers generate the ROC curve along the diagonal (FPR = TPR). The closer the curve is to the 45-degree diagonal in the ROC space, the less accurate the test. The accuracy of a classifier is tested by finding the area under the ROC (AUC) curve as it gives the ratio of TPR and FPR. The ROC (Receiver Operating Characteristic) curve for the test data for four classifiers SVM, KNN, Logistic Regression, and ANN is shown in Figures 2–14. The performance of the classifiers with the hybrid optimization method BGWOPSO is better than with BGWO and PSO. Among the classifiers, ANN performed the best, with an AUC of 1.0, which is the highest value. Figures 2–4 show the ROC curve of Logistic Regression with BGWO, PSO, and BGWOPSO; Figures 5–7 show the ROC curve of SVM with BGWO, PSO, and BGWOPSO; Figures 8–10 show the ROC curve of KNN with BGWO, PSO, and BGWOPSO; Figures 11–13 show the ROC curve of ANN with BGWO, PSO, and BGWOPSO; Figure 14 shows a combined ROC curve for all classifiers with BGWOPSO.

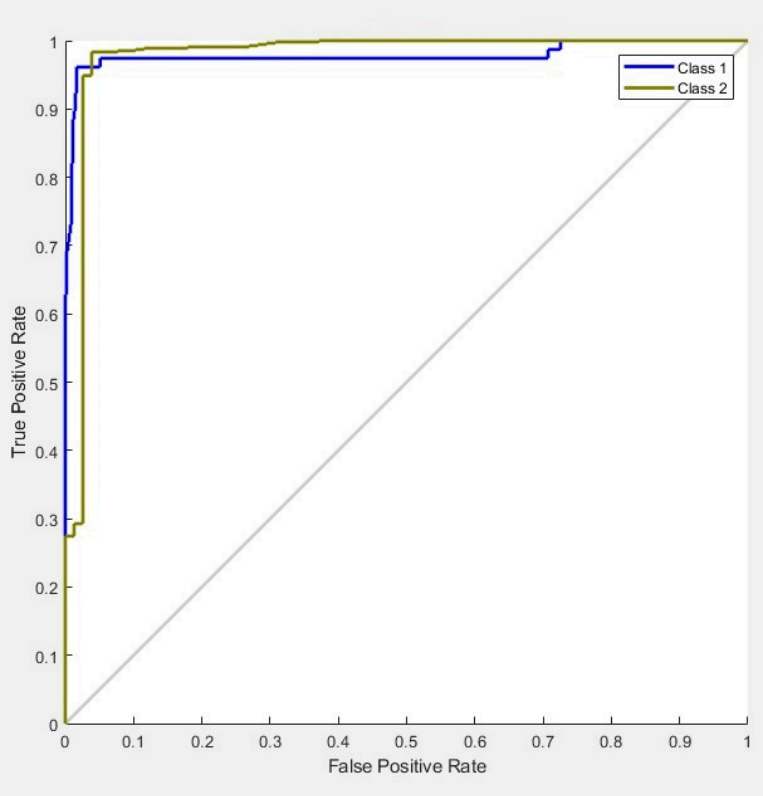

Figure 2. ROC of Logistic Regression with BGWO.

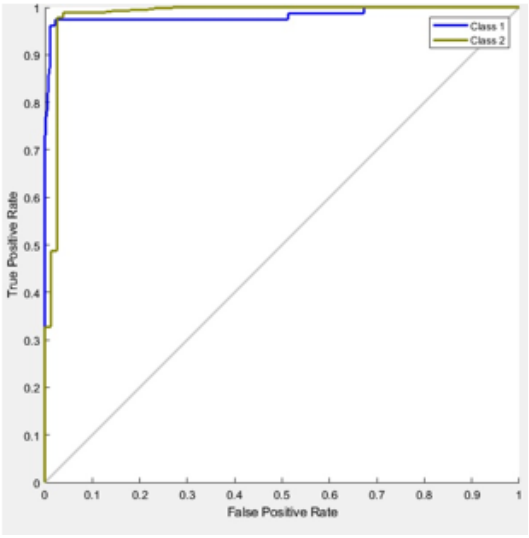

Figure 3. ROC of Logistic Regression with PSO.

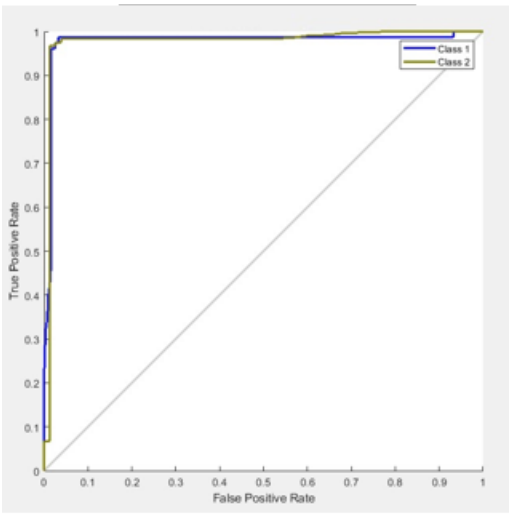

Figure 4. ROC of Logistic Regression with BGWOPSO.

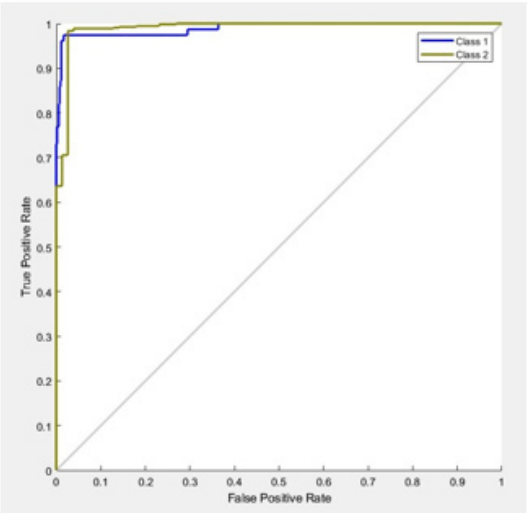

Figure 5. ROC of SVM with BGWO.

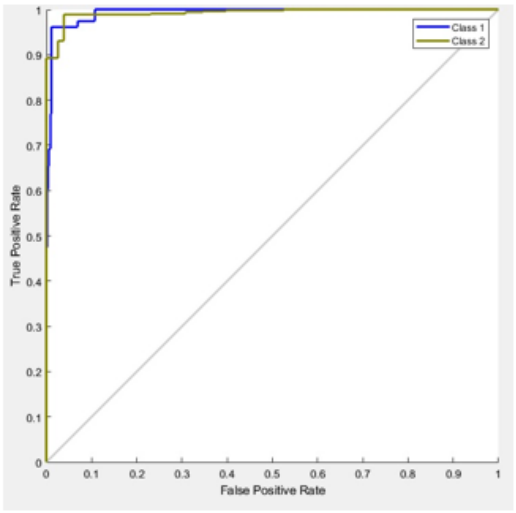

Figure 6. ROC of SVM with PSO.

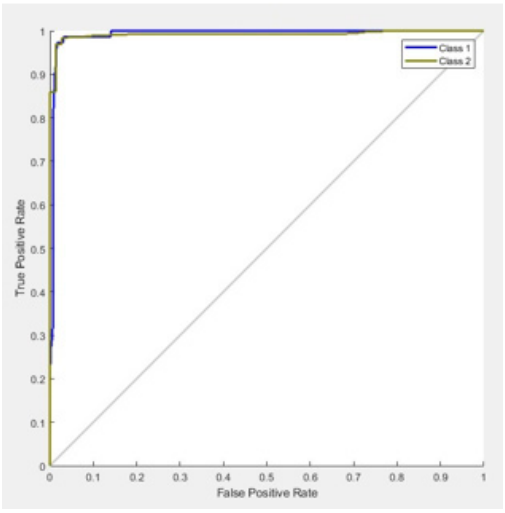

Figure 7. ROC of SVM with BGWOPSO.

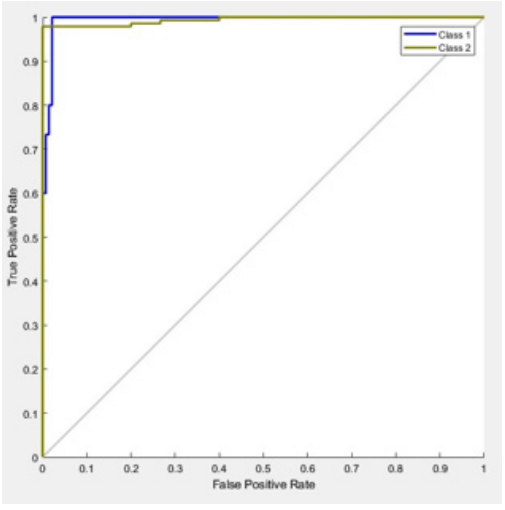

Figure 8. ROC of KNN with BGWO.

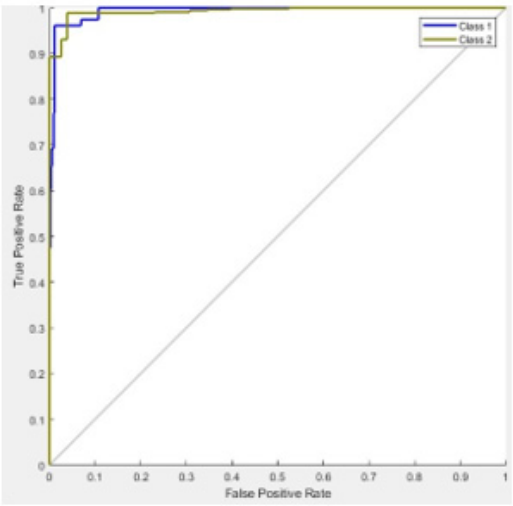

Figure 9. ROC of KNN with PSO.

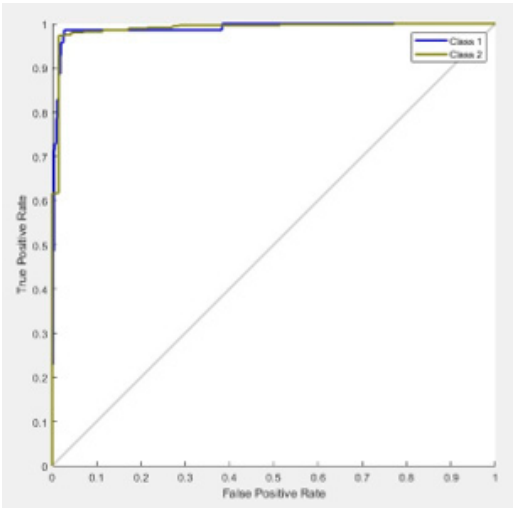

Figure 10. ROC of KNN with BGWOPSO.

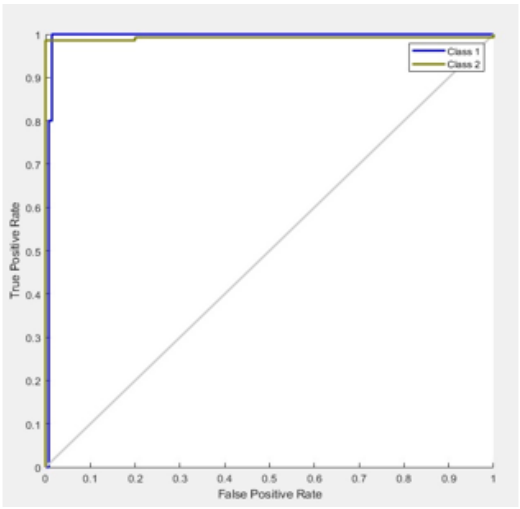

Figure 11. ROC of ANN with BGWO.

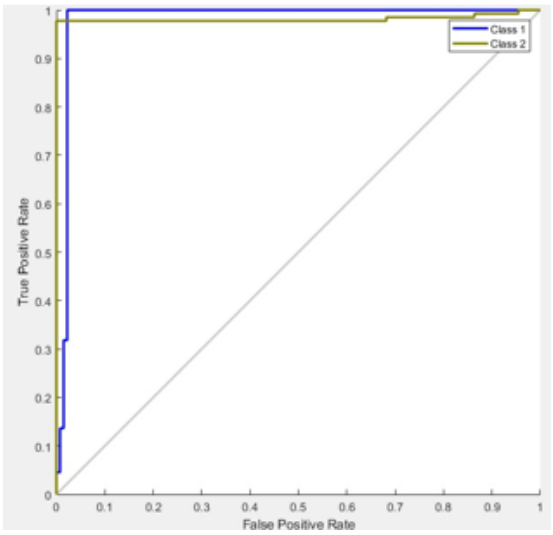

Figure 12. ROC of ANN with PSO.

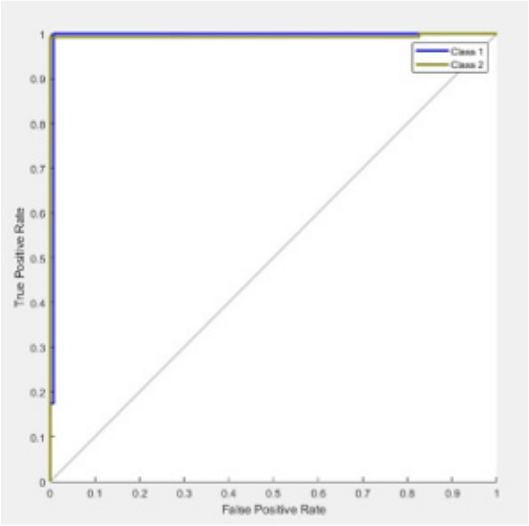

Figure 13. ROC of ANN with BGWOPSO.

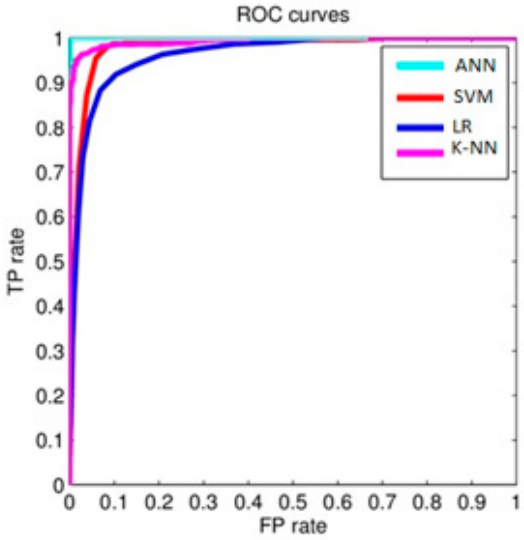

Figure 14. ROC for all classifiers with BGWOPSO.

4.2. Discussion

In this section, the proposed method is compared and evaluated with the approaches presented in [3]. The same dataset was used in this research as in [3], and a comparison is made on the accuracy of the models. The accuracy parameter is for calculating the efficiency of the classification algorithms, which shows the percentage of the entire set of test records that is correctly classified by the model. The proposed hybrid optimization algorithm BGWOPSO obtained the highest accuracy with the ANN classifier in detecting fake accounts on Instagram, with an accuracy of 99.1% as compared to the accuracy of 91.76% achieved by [3]. Figure 15 shows a graphical representation of the accuracy of fake account detection in the Instagram dataset in terms of bar charts. The chart shows values for four different training–testing ratios (60:40, 70:30, 80:20, and 90:10) of four different classifiers KNN, SVM, ANN, and LR, where the *X*-axis is represented with four different classifiers and the *y*-axis is the accuracy in percentage.

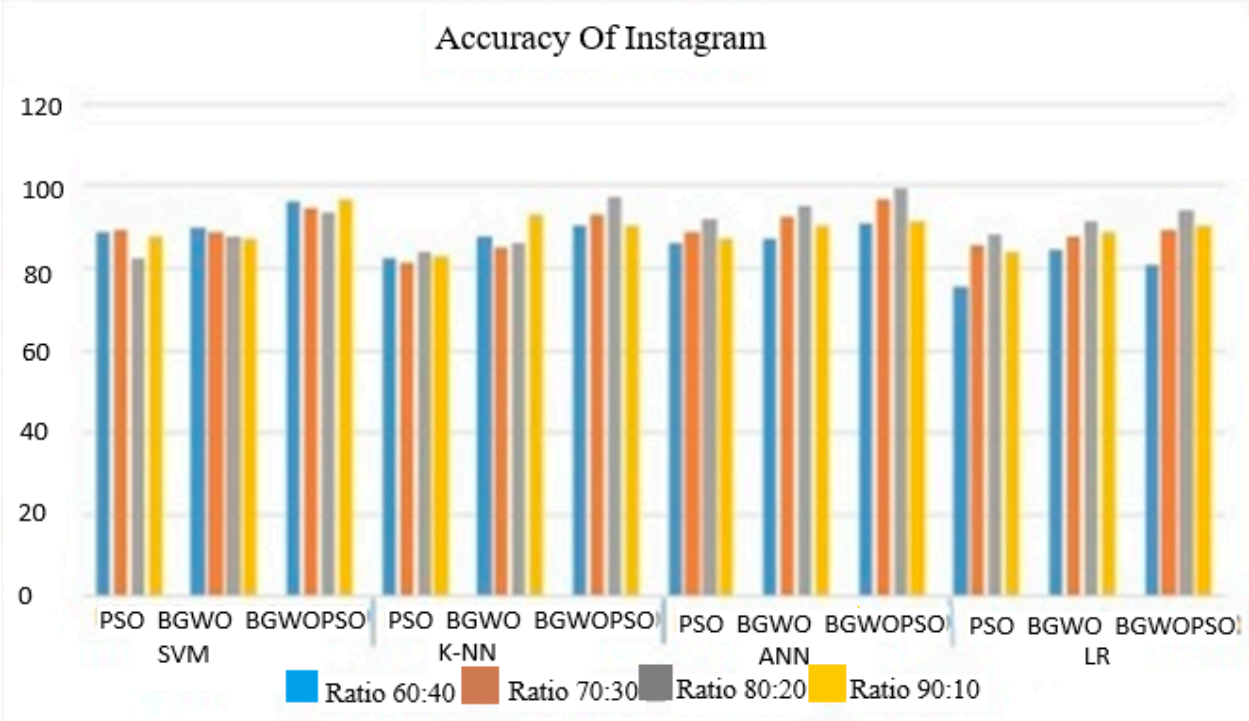

Figure 15. Accuracy of the classifiers with the optimization methods.

Table 4 gives a comparison with the state of the art in detecting fake accounts on Instagram datasets, as given in the literature review in Table 1.

As can be observed from Table 4, the proposed method of hybrid Binary Grey Wolf Optimization with Particle Swarm Optimization when applied to the classifiers gives improved performance. The datasets used in the state of the art are obtained differently and a complete comparison cannot be made. However, the dataset used by Purba, Asirvatham, and Murugesan [3] is the same as used in this research and it can be observed that there is considerable improvement in performance by applying optimization techniques, especially BGWOPSO. Optimization methods have been applied and tested by a number of researchers on Online Social Network (OSN) data and a higher accuracy has been seen in such methods, as shown in the literature review in Table 1.

Table 4. Comparison of BGWOPSO with the state-of-the-art literature.

The State-of-the-Art Literature	Method	Accuracy	AUC
Purba, Asirvatham, and Murugesan [3]	Random Forest	91.76%	
Dey et al. [4]	Random Forest	92.5%	
Akyon and Esat Kalfaoglu [31]	Neural Network with genetic algorithm as feature selection	96%	
Saranya Shree, Subhiksha, and Subhashini [6]	CNN	91.5%	
Meshram et al. [7]	Random Forest	96.94%	
Sheikhi [8]	Bagged Decision Tree	98.45%	
Durga and Sudhakar [32]	Decision Tree	96%	
BGWOPSO	Artificial Neural Network	99.1%	1.0

5. Conclusions and Future Work

Today's world has been likened to a global village due to the existence of the global Internet network. Humans are making new tools every day to communicate with each other faster. One of these tools is social networks, which are becoming more widespread day by day. Despite the many advantages of these networks, they also have risks for users, and one of these risks is the falsification of their names in social networks, which can cause irreparable financial and credit losses to the people in the network.

In this paper, the methods of detecting fake accounts in social networks have been investigated. By examining these methods, it can be seen that machine learning methods have been very effective in detecting fake accounts. For this purpose, in this study, we used machine learning to detect fake accounts. To implement and extract the results, a database of real and fake Instagram accounts available to the public on the Kaggle site has been used. The effective features in detecting fake accounts were selected using Particle Swarm Optimization (PSO), Binary Grey Wolf Optimization (BGWO), and a hybrid algorithm with PSO and BGWO (BGWOPSO). They were evaluated with four classifiers, namely KNN, SVM, ANN, and LR. Finally, the hybrid optimization BGWOPSO achieved the highest accuracy of 99.1% and an AUC of 1.0. The proposed method is a high-precision method that was implemented in the MATLAB simulator.

This research focuses on providing a high-accuracy fake account detection system, and some suggestions that can be made to complete and improve it in the future are mentioned below. As a combination method, the combination of the genetic algorithm and Ant Colony algorithm can be used for detecting fake accounts. As it was said, in the proposed method, high accuracy is preferred. In future work, the time criterion for detecting can be used to improve the identification system, and other evolutionary algorithms can also be used to test the models.

Author Contributions: Conceptualization, P.A. and K.P.; methodology, P.A. and K.P.; software, P.A.; validation, P.A. and K.P.; formal analysis, P.A.; investigation, P.A.; resources, P.A.; data curation, P.A.; writing—original draft preparation, P.A.; writing—review and editing, K.P.; visualization, P.A.; supervision, K.P.; project administration, K.P.; funding acquisition, K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available in kaggle.com (accessed on 19 September 2024).

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

References

1. Hegde, P.; Saurabh, N.; Salian, P. Detection and Classification of Genuine User Profile Based on Machine Learning Techniques. In Proceedings of the 2022 International Conference on Intelligent Technologies (CONIT), Hubli, India, 24–26 June 2022. [CrossRef]
2. Kaubiyal, J.; Jain, A.K. A Feature Based Approach to Detect Fake Profiles in Twitter. In Proceedings of the 3rd International Conference on Big Data and Internet of Things, in BDIOT 2019, Melbourne, Australia, 22–24 August 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 135–139. [CrossRef]
3. Purba, K.; Asirvatham, D.; Murugesan, R.K. Classification of instagram fake users using supervised machine learning algorithms. *Int. J. Electr. Comput. Eng. (IJECE)* **2020**, *10*, 2763–2772. [CrossRef]
4. Dey, A.; Reddy, H.; Dey, M.; Sinha, N. Detection of Fake Accounts in Instagram Using Machine Learning. *Int. J. Comput. Sci. Inf. Technol.* **2019**, *11*, 83–90. [CrossRef]
5. Efthimion, P.G.; Payne, S.; Proferes, N. Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots. *SMU Data Sci. Rev.* **2018**, *1*, 5.
6. Saranya Shree, S.; Subhiksha, C.; Subhashini, R. Prediction of Fake Instagram Profiles Using Machine Learning. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3802584 (accessed on 19 September 2024).
7. Meshram, E.P.; Bhambulkar, R.; Pokale, P.; Kharbikar, K.; Awachat, A. Automatic Detection of Fake Profile Using Machine Learning on Instagram. *Int. J. Sci. Res. Sci. Technol.* **2021**, *8*, 117–127. [CrossRef]
8. Sheikhi, S. An Efficient Method for Detection of Fake Accounts on the Instagram Platform. *Rev. D'intelligence Artif.* **2020**, *34*, 429–436. [CrossRef]
9. Jain, M.; Saihjpal, V.; Singh, N.; Singh, S.B. An Overview of Variants and Advancements of PSO Algorithm. *Appl. Sci.* **2022**, *12*, 8392. [CrossRef]
10. Li, Y.; Lin, X.; Liu, J. An Improved Gray Wolf Optimization Algorithm to Solve Engineering Problems. *Sustainability* **2021**, *13*, 3208. [CrossRef]
11. Pellet, H.; Shiaeles, S.; Stavrou, S. Localising social network users and profiling their movement. *Comput. Secur.* **2019**, *81*, 49–57. [CrossRef]
12. Egele, M.; Stringhini, G.; Kruegel, C.; Vigna, G. Towards Detecting Compromised Accounts on Social Networks. *IEEE Trans. Dependable Secur. Comput.* **2017**, *14*, 447–460. [CrossRef]
13. Agarwal, N.; Jabin, S.; Hussain, S.Z. Analyzing Real and Fake users in Facebook Network based on Emotions. In Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS), Bangalore, India, 7–11 January 2019; pp. 110–117. [CrossRef]
14. Chen, B.; Xiong, Z.; Zhao, Y.; Zhang, J. Transformation of Mg-Bearing Minerals and its Effect on Slagging During the High-Alkali Coal Combustion. Available online: https://ssrn.com/abstract=4941607 (accessed on 19 September 2024).
15. Khaled, S.; El-Tazi, N.; Mokhtar, H.M.O. Detecting Fake Accounts on Social Media. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 3672–3681. [CrossRef]
16. Kondeti, P.; Yerramreddy, L.P.; Pradhan, A.; Swain, G. Fake Account Detection Using Machine Learning. In *Evolutionary Computing and Mobile Sustainable Networks*; Suma, V., Bouhmala, N., Wang, H., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer: Singapore, 2021; pp. 791–802. [CrossRef]
17. Suganya, R.; Muthulakshmi, S.; Venmuhilan, B.; Kumar, K.V.V.; Vignesh, G. Detect Fake Identities Using Improved Machine Learning Algorithm. Undefined. 2021. Available online: https://www.semanticscholar.org/paper/Detect-fake-identities-using-improved-Machine-Suganya-Muthulakshmi/4b4e968545cb233b351249c2cee884be37fcf0bc (accessed on 26 September 2022).
18. Bindu, P.V.; Mishra, R.; Thilagam, P.S. Discovering spammer communities in twitter. *J. Intell. Inf. Syst.* **2018**, *51*, 503–527. [CrossRef]
19. Patil, A.P.; Remulkar, V.; Hardik; Shirole, U. Social Networks Fake Detection. *Int. J. Recent Adv. Multidiscip. Top.* **2022**, *3*, 98–100.
20. Prabhu Kavin, B.; Karki, S.; Hemalatha, S.; Singh, D.; Vijayalakshmi, R.; Thangamani, M.; Haleem, S.L.A.; Jose, D.; Tirth, V.; Kshirsagar, P.R.; et al. Machine Learning-Based Secure Data Acquisition for Fake Accounts Detection in Future Mobile Communication Networks. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, e6356152. [CrossRef]
21. Kadhim, A.; Abdullah, A. Fake accounts detection on social media using stack ensemble system. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 3013–3022. [CrossRef]
22. Benabbou, F.; Boukhouima, H.; Sael, N. Fake accounts detection system based on bidirectional gated recurrent unit neural network. *Int. J. Electr. Comput. Eng. IJECE* **2022**, *12*, 3129. [CrossRef]
23. David, I.; Siordia, O.S.; Moctezuma, D. Features combination for the detection of malicious Twitter accounts. In Proceedings of the 2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 9–11 November 2016; pp. 1–6. [CrossRef]
24. Sowmya, P.; Chatterjee, M. Detection of Fake and Cloned Profiles in Online Social Networks. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3349673 (accessed on 19 September 2024).
25. Bharti, K.K.; Pandey, S. Fake account detection in twitter using logistic regression with particle swarm optimization. *Soft Comput.* **2021**, *25*, 11333–11345. [CrossRef]
26. Homsi, A.; Al-Nemri, J.; Naimat, N.; Kareem, H.; Al-Fayoumi, M.; Snober, M.A. Detecting Twitter Fake Accounts using Machine Learning and Data Reduction Techniques. In Proceedings of the 10th International Conference on Data Science, Technology and Applications (DATA 2021), Online, 6–8 July 2021; pp. 88–95. [CrossRef]

27. SudalaiMuthu, T.; Reddy, C.D.K.; Reddy, B.S.; Sahithya, M.L.; Visalaxi, S. Detecting spammer and fake user on social networks using machine learning approach. *AIP Conf. Proc.* **2022**, *2385*, 050010. [CrossRef]
28. Awan, M.J.; Khan, M.A.; Ansari, Z.K.; Yasin, A.; Shehzad, H.M.F. Fake profile recognition using big data analytics in social media platforms. *Int. J. Comput. Appl. Technol.* **2022**, *68*, 215–222. [CrossRef]
29. Munga, J.B.; Mohandas, P. Feature Selection for Identification of Fake Profiles on Facebook. In Proceedings of the 6th Kuala Lumpur International Conference on Biomedical Engineering 2021, Online, 28–29 July 2021; Usman, J., Liew, Y.M., Ahmad, M.Y., Ibrahim, F., Eds.; IFMBE Proceedings; Springer International Publishing: Cham, Switzerland, 2022; pp. 489–497. [CrossRef]
30. Gupta, A.; Kaushal, R. Towards detecting fake user accounts in facebook. In Proceedings of the 2017 ISEA Asia Security and Privacy (ISEASP), Surat, India, 29 January–1 February 2017; pp. 1–6. [CrossRef]
31. Akyon, F.C.; Kalfaoglu, M.E. Instagram Fake and Automated Account Detection. In Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 31 October–2 November 2019; pp. 1–7. [CrossRef]
32. Durga, P.; Sudhakar, D.T. The use of supervised machine learning classifiers for the detection of fake instagram accounts. *J. Pharm. Negat. Results* **2023**, *14*, 267–279. [CrossRef]
33. My Information Bubble Project. Available online: http://mib.projects.iit.cnr.it/ (accessed on 29 March 2024).
34. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
35. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
36. Talbi, E.-G. A Taxonomy of Hybrid Metaheuristics. *J. Heuristic* **2022**, *8*, 541–564. [CrossRef]
37. Singh, N.; Singh, S.B. Hybrid algorithm of particle swarm optimization and Grey Wolf optimizer for improving convergence performance. *J. Appl. Math.* **2017**, *2017*, 2030489. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.