EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Grasshopper-Based Detection of Fake Social Media Profiles

Nadir Mahammed[1*], Imène Saidi[1], Khayra Bencherif[1], Miloud Khaldi[1], Mahmoud Fahsi[2], Zouaoui Guellil[3]

[1]LabRI-SBA Lab. Ecole Supérieure en Informatique Sidi Bel Abbès, Algeria
[2]EEDIS Lab. Djillali Liabes University Sidi Bel Abbes, Algeria
[3]LabRI-SBA Lab. Hassiba BenBouali University Chlef, Algeria

Abstract

The proliferation of fake profiles on social media platforms presents a growing challenge for digital ecosystems, where the detection of such profiles is critical to maintaining the integrity of online environments. This paper introduces a hybrid approach that integrates the Grasshopper Optimization Algorithm with various Machine Learning classifiers, including Support Vector Machine, Naive Bayes, and Random Forest. The nature-inspired metaheurisitic used is employed to optimize key hyperparameters of these classifiers, thereby enhancing their performance in detecting fake profiles. The proposed method is evaluated on a well defined balanced dataset, demonstrating significant improvements in classification performance, particularly in terms of accuracy, precision, recall, and F1-score. The results suggest that the proposed hybrid approach can effectively address the challenges associated with balanced and imbalanced datasets in fake profile detection. Furthermore, the study discusses potential directions for improving scalability and applying the approach to larger and more dynamic datasets in the future.

Received on 09 January 2025; accepted on 22 July 2025; published on 24 July 2025

Keywords: Online social network, fake profiles detection, nature-inspired algorithm, grasshopper optimization algorithm, machine learning

doi:10.4108/eetsis.7159

1. Introduction

In today's digital world, social media platforms have become the modern equivalents of town squares, enabling unparalleled levels of connection and information exchange. However, this democratization of voice comes with a significant downside: the rampant spread of fake profiles [1]. These fabricated identities, often orchestrated by bots or malicious actors, pose a serious threat to the integrity of online discourse. They act as vectors for the dissemination of misinformation, manipulation of public opinion, and the proliferation of cybercrime. A recent study by [2] suggests that up to 15% of social media accounts could be bots, highlighting the vast scale of this issue.

Identifying these digital frauds is a complex challenge. fake profiles often mimic human behavior patterns, leverage advanced language models, and even hijack legitimate user credentials [3]. Traditional rule-based detection systems struggle in this dynamic environment, as adversaries continuously adapt their tactics. Machine learning (ML) offers a more adaptable solution, capable of learning patterns that elude rule-based systems. However, even ML approaches face obstacles. Datasets for training these models are often limited and small, as manual labeling is a time-intensive and resource-intensive process [4]. Additionally, the feature space used for analysis can be high-dimensional [5], encompassing factors like user activity, network structures, and textual content. Furthermore, the decision boundaries separating real and fake profiles are likely to be highly non-linear, making classification even more challenging.

*Corresponding author. Email: n.mahammed@esi-sba.dz

Recent advancements in nature-inspired metaheuristic algorithms offer promising solutions. These algorithms, which draw inspiration from natural phenomena, have demonstrated remarkable capabilities in navigating complex problem spaces [6]. One such algorithm, the Grasshopper Optimization Algorithm (GOA) [7], mimics the social behavior of grasshoppers, including their attraction-repulsion dynamics and wind-guided movements. While GOA has achieved success in various optimization tasks, its potential application in ML, particularly for high-stakes classification problems like fake profiles detection, remains largely unexplored.

This paper aims to bridge this gap. The authors propose a novel approach to fake profiles detection that leverages the optimization power of GOA in conjunction with a specific Fitness function. The core hypothesis rests on two pillars:

- GOA's nature-inspired dynamics can effectively navigate the feature space of social media data, potentially surpassing traditional optimizers, even when paired with a simple ML base model.
- Despite its perceived complexity, the problem of fake profiles detection may contain more linearly separable components than previously assumed by the ML community.

The paper is structured into six sections. Section 2 presents the current state of the art regarding the detection of fake profiles. Section 3 outlines the materials and methodology utilized in this study, including details on the dataset, preprocessing techniques, the proposed approach, and its characteristics. Following this, Section 4 presents the performance metrics obtained from experimental results and provides a discussion of these findings. Finally, Section 5 concludes the paper.

2. Related work

This section reviews various approaches for identifying fake profiles on social media sites, highlighting the use of numerous machine learning algorithms.

The article [8] proposed a framework for detecting fake profiles on social media. This framework leveraged open-source big data tools and utilized Long Short-Term Memory (LSTM) networks for analysis, employing the Dispersive Flies Optimization (DFO) metaheuristic for improved feature selection from the dataset. The approach emphasizes ethical data collection, considering both public and private user attributes.

In [9], the authors addressed fake profiles detection on Facebook with a hybrid approach. This two-stage process involves identifying initial clusters using the Satin Bowerbird Optimization Algorithm (SBO) to find the best centroids for classifying profiles as real or fake. Subsequently, the K-Means clustering algorithm is employed to classify each profile as real or fake.

In [10], the study addresses a gap by comparing various detection techniques, showing promise for both supervised and unsupervised ML models, and highlighting the potential of bio-inspired algorithms for superior fake profiles detection on online social networks. The authors proposed adapting the Grey Wolf Optimizer Algorithm (GWO), a swarm-inspired algorithm, from natural inspiration to artificial implementation for detecting fake profiles on social networks.

The authors of [11] proposed using a bio-inspired algorithm called the Fire Hawk Optimizer (FHO) to address the challenge of detecting fake profiles. They tested different feature groups from a Twitter dataset to determine which ones are most effective for spotting fake profiles. They identified and proposed using the Gradient Boosting Classifier (GBC) as the Fitness function.

In [12], the paper addressed fake profiles on social media. The authors proposed a new method that combines multiple ML algorithms to analyze user behavior and profile details. This "ensemble" approach uses a Majority Voting Technique (MVT) to determine if a profile is fake or real. The results suggest that their method is promising for enhancing the safety of social media platforms.

The authors of [13] explored the use of ML to combat fake profiles on social media. They tested a variety of ML algorithms to identify which ones are most effective at detecting these fake accounts. The paper emphasizes the importance of using different evaluation techniques, such as confusion matrices and various error rate calculations, to select the best-performing ML model.

In [14], the study investigated the effectiveness of the Fire Hawk Optimizer (FHO) for detecting fake social media profiles. The authors evaluated FHO's performance using a Gradient Boosting Classifier as the Fitness function, aiming to identify the optimal feature subsets from a Facebook dataset that best distinguish between genuine and fake profiles.

The authors of [15] proposed a new approach that employs a rich feature set to analyze profile information, network connections, and user behavior to identify fake profiles. They also introduced an optimizable Bagged Tree Algorithm (BTA), which builds a robust decision tree model by removing irrelevant branches, leading to improved Accuracy and efficiency.

The paper [16] proposed a hybrid machine learning model using logistic regression (LR) and gradient descent optimization (GBO) to detect fake profiles on online social networks. The model is evaluated on Instagram data, with a dataset of 7,500 accounts. It achieves 92.70% Accuracy, outperforming other classifiers like SVM and decision trees.

In [17], they proposed a comprehensive solution framework to detect and eradicate fake profiles

on social media. It leverages ML algorithms to analyze various aspects of user behavior and identify anomalies associated. It emphasizes user education, continuous updates, and privacy-respecting measures, by employing algorithms like ANN, RF, Extreme Gradient Boost (XGBoost), LSTM, CNN, and a Voting Classifier (VC).

[18] used a multitask BERT model for sentiment and topic classification, highlighting the effectiveness of transformer-based approaches in complex text analysis tasks relevant to fraud detection.

A recent study on Bitcoin price prediction employed the N-BEATS deep learning model to capture complex temporal patterns, showing the adaptability of advanced ML in volatile, fraud-prone environments [19].

A recent study, EcomFraudEX [20], proposed an explainable ML framework using ensemble models and feature selection to classify e-commerce fraud. Though focused on e-commerce, its use of interpretable ensemble learning is relevant to fraud detection in social platforms.

Table 1 provides a comprehensive overview of recent advancements in fake profiles detection across various online social networks (OSNs).

The majority of the studies rely on ML algorithms for fake profiles detection, achieving high Accuracy rates exceeding 92%. This suggests that ML approaches are effective in identifying fake profiles based on user data. Three studies ([9], [11], and [14]) explore metaheuristics such as SBO and FHO. While these methods achieve impressive results (98% - 99%), further research is needed to understand their broader applicability compared to established ML techniques.

Popular choices include k-means clustering, Gradient Boosting Classifier, Random Forest, and multivariate techniques. These algorithms excel at classifying users as real or fake based on extracted features. One study ([8]) utilizes Long Short-Term Memory (LSTM) networks, demonstrating their potential for fake profiles detection using social media content (text or sequence data).

Other considerations include studies that combine ML with additional techniques, such as SBO and FHO, for potentially enhanced performance ([9], [11], and [14]).

Table 1 covers studies on Facebook, Twitter, and Instagram, indicating that the effectiveness of detection methods may vary depending on the platform's specific characteristics, such as user behavior and data availability. The dataset sizes vary across studies (1244 - 17,350 entries). While larger datasets are generally preferred for robust model training, some studies achieve high Accuracy even with smaller datasets. This suggests that selecting the appropriate technique is crucial for optimal performance.

Current research on fake profiles detection in social networks heavily relies on ML techniques, which have achieved significant success. However, bio-inspired algorithms remain largely unexplored in this domain. This work proposes a novel approach that leverages a combination strategy, incorporating a bio-inspired algorithm for identifying fake social media profiles.

3. Materials and methods

This section begins with a detailed examination of the dataset used in this study. The preprocessing techniques applied to prepare the data for analysis will then be summarized. Following this, a concise overview of the various ML algorithms considered for this task will be presented. Finally, the core element of this work, the bio-inspired algorithm, will be introduced. A brief description of the algorithm's design principles, the rationale for its hybridization, and its operational details will be provided.

3.1. Dataset

This study utilizes a balanced dataset sourced from the online social network Twitter [21]. The dataset contains 16 features, as shown in Table 3. It comprises a total of 1,000 social media profiles, meticulously curated to achieve a balance between fake accounts (501) and real accounts (499). This balanced representation of classes ensures optimal results for the subsequent analysis (see Table 2).

3.2. Dataset preprocessing

The cornerstone of successful data analysis lies in meticulous data preprocessing [22]. This crucial preliminary stage transforms raw data, which is often riddled with inherent noise, inconsistencies, missing values, and format variations, into a comprehensible format [23]. Data preprocessing addresses these challenges, ensuring that the data is well-conditioned for subsequent analysis. In the domain of ML specifically, this process involves applying algorithms that require structured and clean data. By thoroughly addressing these data quality issues, data preprocessing techniques pave the way for robust statistical modeling and algorithm implementation [24].

Figure 1 presents a detailed overview of the 12-step data preprocessing sequence employed in this study [24]. This process meticulously transforms the raw Twitter data into a well-structured format, suitable for subsequent analysis.

- Step 1: Text Normalization: Removes extraneous elements such as Unicode strings, URLs, user mentions, and hashtag symbols, focusing the analysis on the core textual content.

Table 1. Related work summary.

Reference	OSN	ML	Metaheuristic	Other	Dataset Size	Results (Acc)
[8]	Facebook	LSTM	DFO	–	–	0.979
[9]	Facebook	k-means	SBO	Hybridization	1244	0.989
[10]	Facebook	–	GWO	Transition	1244	0.980
[11]	Twitter	GBC	FHO	Hybridization	17350	0.996
[12]	Twitter	MVT	–	Combination	6825	0.991
[13]	Instagram	RF	–	Comparison	6868	0.997
[14]	Facebook	GBC	FHO	Hybridization	1244	0.998
[15]	Facebook	BTA	–	–	–	0.999
[16]	Instagram	LR, GBO	–	Combination	7500	0.927
[17]	Twitter	XGBoost, VC	–	Comparison	6825	0.991

Table 2. Twitter profile dataset

	Real	Fake
Record	499	501
Proportion (%)	49.9	50.1

- Steps 2 & 3: Sentiment Augmentation: Emoticons and emojis are systematically mapped to their corresponding word meanings, enriching the data with sentiment information. Abbreviations and acronyms are consistently expanded to improve data clarity and reduce ambiguity.
- Step 4: Error Correction: Spelling mistakes are meticulously corrected to ensure data integrity and facilitate accurate analysis.
- Steps 5 & 6: Text Normalization: Contractions are systematically expanded to preserve intended meaning while enhancing data readability. Elongated characters are judiciously abbreviated for concise representation.
- Steps 7-11: Feature Engineering:
  - Punctuation is systematically removed, streamlining the text for analysis.
  - Case folding is applied to convert all text to lowercase, ensuring consistency and simplifying subsequent processing steps.
  - Word segmentation meticulously separates individual words to facilitate analysis.
  - Numbers are intentionally removed, as they might not be relevant for the specific research focus.
  - Stop words, frequently occurring words with minimal semantic value, are strategically removed to enhance the signal-to-noise ratio in the data.

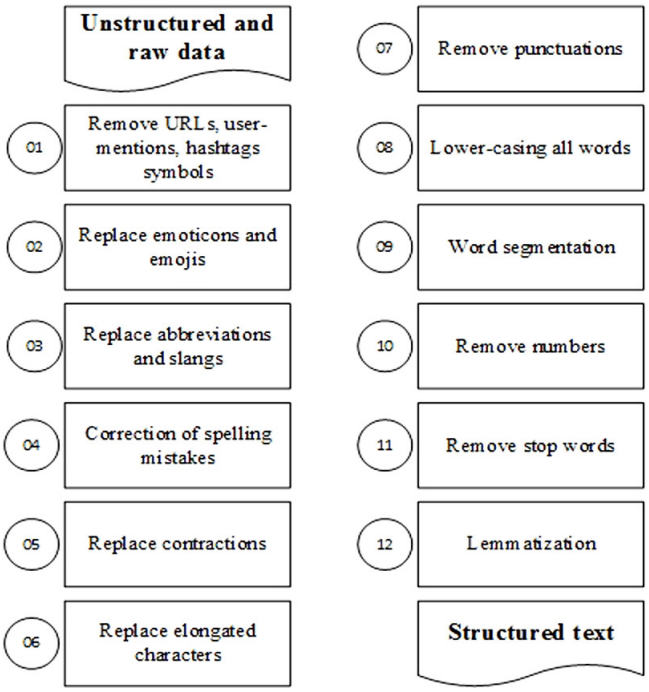

Figure 1. Preprocessing combination

- Step 12: Lexical Normalization: Lemmatization is meticulously applied to reduce words to their base forms, promoting data compactness and facilitating effective analysis.

3.3. Machine learning algorithms

Equipped with the preprocessed Twitter data, this section briefly presents the ML techniques employed in this work.

Induction of Decision tree (ID3) Algorithm. The ID3 algorithm is a supervised learning technique that constructs a tree-like structure based on the knowledge

Table 3. Twitter dataset attributes

Attribute Name	Description
describing_account	Length of the user-defined string describing the account.
protected	When true, indicates that this user has chosen to protect their Tweets.
followers_count	The number of followers this account currently has.
friends_count	The number of users this account is following.
statuses_count	The number of Tweets (including retweets) issued by the user.
favourites_count	The number of Tweets this user has liked in the account's lifetime.
listed_count	The number of public lists that this user is a member of.
verified	When true, indicates that the user has a verified account.
profile_use_background_image	When true, indicates the user wants their uploaded background image to be used.
contributors_enabled	Indicates the user has an account with "contributor mode" enabled.
default_profile	When true, indicates the user has not altered the theme or background of their profile.
default_profile_image	When true, indicates the user has not uploaded their own profile image and a default image is applied.
is_translator	When true, indicates the user is a participant in Twitter's translator community.
hashtags_average	Number of hashtags the user has used in their last 20 tweets.
mentions_average	Number of mentions the user has used in their last 20 tweets.
urls_average	Number of URLs the user has included in their last 20 tweets.

extracted from the training data. This structure is used to categorize unseen data points [25].

K-means algorithm. This widely recognized unsupervised learning approach is used for data clustering. The K-means algorithm iteratively groups data points into a predefined number of clusters based on their similarity [26].

K-nearest neighbors classification (K-NN). This supervised learning algorithm leverages the principle of proximity, classifying data points based on the labels of their closest neighbors in the training data [27].

Naive Bayes classifier (NB). A mainstay in supervised learning, the NB employs Bayes' theorem for classification. It relies on the assumption of conditionally independent attributes given the class, which grants NB remarkable computational efficiency and often surprisingly good results [28].

Random forest algorithm (RF). The Random Forest algorithm is a robust ensemble learning technique that leverages multiple diverse decision trees constructed from random subsets of the training data. Each individual tree operates independently, contributing its unique perspective to the overall classification or regression task [29].

Support vector machine (SVM). The Support Vector Machine is a powerful tool for constructing robust classifiers. Its primary objective is to establish an optimal decision boundary within a feature space, effectively separating data points belonging to distinct classes. This decision boundary allows the SVM to predict the class labels of unseen data points [30].

3.4. Proposed algorithm

This section presents a detailed exploration of the Grasshopper Optimization Algorithm [7]. It begins with an examination of the algorithm's core mechanisms, followed by a comprehensive analysis of the Fitness function employed for performance evaluation. Subsequently, it investigates the scientific basis for selecting GOA in this context. This investigation entails a meticulous analysis of the bio-inspired design principles underlying the algorithm. By establishing a clear link between these biological phenomena and their translation into the optimization framework, the section justifies the suitability of GOA for the problem at hand.

Grasshopper optimization algorithm. GOA is a nature-inspired metaheuristic that leverages swarm intelligence principles. Inspired by the natural swarming behavior of grasshoppers, GOA employs a population of candidate solutions, each represented as a "grasshopper" within the search space [7]. The Fitness function serves as the guiding principle, dynamically determining the leader (target) within the swarm based on the performance of each grasshopper (solution). This dynamic leadership mechanism encourages other grasshoppers to move closer to the leader, mimicking the tendency of real grasshoppers to congregate in areas with abundant resources.

Beyond exploitation and exploration, inherent features of many nature-inspired algorithms, GOA incorporates a foraging behavior inspired by the natural food-seeking behavior of grasshoppers. This foraging tendency promotes a more effective search within the solution space [28].

Progress and functioning. Figure 2 presents the pseudo-code for the GOA algorithm.

Algorithm Grasshopper Optimization Algorithm (GOA)

Parameters initialization: c_{Max}, c_{Min}, $Max_{Iteration}$
Population initialization: $X_i (i = 1, 2, ..., t)$
Calculate each individual fitness value
Assign K to the highest fitness value individual
while $k = 1$ to $Max_{Iteration}$ **do**
  Update c_i for each individual using Eq. 1
  for each individual in the population **do**
    Normalize the distances between individuals into [1, 4]
    Update the position of the individual using Eq. 2
    if the individual exceeds the boundaries **then** bring it back
  end for
  Re-evaluate each individual fitness
  if there is a better solution **then** replace K with it
end while

Figure 2. GOA pseudo-code

$$c_i = c_{\text{Max}} - \frac{k-1}{\text{Max}_{\text{Iteration}} - 1} (c_{\text{Max}} - c_{\text{Min}}) \tag{1}$$

- c_{Max} and c_{Min} are the upper and lower bounds of the coefficient c, respectively.
- k denotes the current iteration number.
- $\text{Max}_{\text{Iteration}}$ represents the total number of iterations in the algorithm.

$$X_d^i = c_1 \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} c_2 \cdot \frac{ub_d - lb_d}{2} \cdot s\left(\left|x_j^d - x_i^d\right|\right) \cdot \frac{x_j - x_i}{d_{ij}} \right) + \hat{K}_d \tag{2}$$

- Within each dimension d, the upper and lower bounds are denoted by ub_d and lb_d, respectively.
- $\hat{K}_d$ represents the current best solution identified in that dimension.
- The parameter c_1 is analogous to the inertia weight (ω) used in Particle Swarm Optimization (PSO) [31]. It serves to gradually reduce the movement (amplitude) of grasshoppers around the target (food source), thereby balancing intensification (exploitation) and diversification (exploration) in the search process.
- Parameter c_2 is used to progressively decrease the repulsion zone, attraction zone, and comfort zone between grasshoppers as the number of iterations increases.
- Notably, c_1 (see Equation 1) and c_2 are often combined and expressed using a single equation, as shown in Equation 2 [32].

```
def calculate_fitness(X, y, feature_mask):
    selected_features = X[:, feature_mask.astype(bool)]
    model = LogisticRegression(random_state=0)
    model.fit(selected_features, y)
    y_pred = model.predict(selected_features)
    return accuracy_score(y, y_pred)
```

Figure 3. Feature selection function

Fitness Function. GOA implements a Fitness function for feature selection using logistic regression (LR). Here's a breakdown of the Fitness function:

- Feature selection function (`calculate_fitness`): As shown in Figure 3, this function takes features, target values, and a feature mask as input. The feature mask indicates which features are currently selected. Then, it uses a logistic regression model to evaluate the classification performance based on the selected features and returns the Accuracy score.
- GOA (`grasshopper_optimization`): As shown in Figure 4, this function implements the core GOA logic. It takes various parameters such as data, target values, number of grasshoppers, and iteration limits.

A few words about Fitness Function Inference. Instead of using a pre-defined Fitness function, the Fitness value is calculated within the *grasshopper_optimization* function itself. It computes the Accuracy of a logistic regression model trained on the features selected by the current "grasshopper" (candidate solution).

The Accuracy score is inverted $(1/(1 + \text{Accuracy}))$ to convert it into a minimization problem, where lower values represent better solutions. This inverted Accuracy score serves as the Fitness value for the current "grasshopper".

This approach calculates Fitness values based on the inverted classification Accuracy of an LR model trained on the features selected by the "grasshopper".

GOA is compared with various popular ML techniques (ID3, SVM, NB, RF, K-NN with different k values, and K-means). Each technique is run 1000 times. The parameter settings for GOA and the ML algorithms are summarized in Tables 4 and 6, respectively.

4. Results and discussion

Following the presentation of the proposed approach and its theoretical foundation, the evaluation criteria used, the resulting findings, and a discussion of their significance are presented.

```python
def grasshopper_optimization(X, y, n_grasshoppers=10, max_iter=100, alpha=1.0, beta=1.0, lb=None, ub=None):
    # Lower bound of the decision variables
    # Upper bound of the decision variables
    # Initialize the grasshoppers and their positions and fitness values
    # Iterate for the maximum number of iterations
    for t in range(max_iter):
        # Calculate the distance between the grasshoppers
        # Calculate the attraction and repulsion forces between the grasshoppers
        # Update the position of each grasshopper based on the attraction and repulsion forces
        for i in range(n_grasshoppers):
            # Calculate the total force
            # Update the position of the grasshopper
            # Apply the boundary constraints
            # Evaluate the fitness of the grasshopper
    # Select the best grasshopper based on its fitness value
    best_grasshopper = grasshoppers[np.argmax(grasshoppers_fitness)]
    return best_grasshopper
```

Figure 4. Grasshopper optimization function

Table 4. GOA parameters setting

Parameter	Symbol	Value
Number of Grasshoppers	$n_{\text{grasshoppers}}$	10
Maximum Iterations	max_{iter}	100
Alpha	α	1.0
Beta	β	1.0
Lower Bound	lb	-
Upper Bound	ub	-

Table 5. ML parameters setting

Algorithm	Parameter	Value
ID3	Maximum depth	None
	Splitting criterion	Information gain
SVM	Kernel	Radial basis function
	Regularization parameter	1.0
	Gamma	0.066
NB	Smoothing parameter	1
RF	Number of trees	100
	Maximum depth	None
	Bootstrap samples	True
K-NN	Distance metric	Euclidean
	Weights	Uniform
	K	Different
K-means	Number of clusters	2
	Initialization method	Random
	Maximum iterations	300
	Number of initialization	10

4.1. Evaluation Criteria

The evaluation of the classification model relies on a confusion matrix, an $N \times N$ table where N signifies the number of target groups [33]. This matrix captures the correspondence between predicted and true class labels. In this work, the authors utilize *Accuracy* and *Precision* as primary metrics. It is acknowledged that other metrics, such as *F1-score*, and *Recall*, are also relevant for a comprehensive performance assessment.

- TP = True Positive: The number of instances correctly predicted as the positive class.
- FP = False Positive: The number of instances incorrectly predicted as the positive class.
- TN = True Negative: The number of instances correctly predicted as the negative class.
- FN = False Negative: The number of instances incorrectly predicted as the negative class.

4.2. Results and Discussion

The proposed approach achieved remarkable success in identifying fake profiles, as illustrated in the confusion matrix shown in Table 6.

First, the approach demonstrated high Accuracy in profile classification. With a total of 432 *True Positives* (correctly classified real profiles) and 496 *True Negatives* (correctly classified fake profiles), the approach exhibits a strong ability to distinguish between real and fake profiles, resulting in an Accuracy of 93.9%. This indicates a well-performing method.

Second, the approach proved effective in detecting fake profiles. The high number of *True Positives* (432) underscores the method's proficiency in accurately identifying fake profiles.

Third, the approach showed a low rate of misidentifying real profiles. The relatively low number of *False Negatives* (29) signifies that the method minimizes the risk of erroneously classifying real users as fake. This helps ensure a positive user experience by avoiding unnecessary restrictions on legitimate accounts.

Table 6. Confusion matrix

Predicted class	Real positive	Real negative	Total
Positive	TP = 432	FP = 32	465
Negative	FN = 29	TN = 496	525
Total	461	529	1000

Table 7. Obtained results.

Algorithm	Accuracy	Precision	Recall	F1-score
ID3	0.890	0.877	0.882	0.878
SVM	0.844	0.823	0.832	0.827
K-NN (k=3)	0.856	0.837	0.841	0.834
NB	0.889	0.874	0.880	0.887
RF	0.892	0.870	0.888	0.878
k-means	0.670	0.654	0.667	0.654
GOA	**0.939**	**0.928**	0.931	0.925

Finally, the approach demonstrated balanced classification performance. The comparable values of *True Positives* (432) and *True Negatives* (496) suggest that the method is not biased towards a specific class. It effectively handles both real and fake profiles with a high degree of Accuracy.

Table 7 (labeled "Obtained Results") presents the performance metrics (*Accuracy*, *Precision*, *Recall*, and *F1-score*) for various classification algorithms applied to the task of identifying fake profiles.

First of all, the overall superiority of GOA is clearly demonstrated. GOA stands out as the top performer, achieving the highest *Accuracy* (93.9%) and *F1-score* (92.5%) among all the algorithms. This indicates that GOA effectively explores the solution space and optimizes the classification approach, leading to superior performance in identifying fake profiles.

Next, the strengths of GOA in balancing exploration and exploitation are demonstrated. GOA's ability to balance these two aspects during optimization likely contributes to its success. Exploration allows the algorithm to discover new and potentially better regions of the solution space, while exploitation focuses on refining promising areas.

Moreover, GOA's design is well-suited for handling complex optimization problems, including those with non-continuous or noisy search spaces, such as fake profiles detection. This makes it a suitable choice for tackling the challenge of classifying fake profiles, which may involve intricate patterns and data characteristics.

Furthermore, the choice of logistic regression as the classification approach enhances the performance. Logistic regression's ability to model the relationship between features and binary outcomes (real or fake profiles) aligns well with the task of identifying fake profiles.

Table 8. Accuracy comparison

Algorithm	[21]	Our proposition
ID3	-	0.890
SVM	-	0.844
K-NN (k=3)	-	0.856
NB	0.909	0.889
RF	-	0.892
k-means	-	0.670
Our approach (GOA)	-	**0.939**

The remaining algorithms (ID3, SVM, K-NN, NB, RF, and K-means) exhibit varying levels of performance. ID3, NB, and RF achieved *Accuracy* in the range of 88.9% – 89.2%, comparable to GOA's 93.9%. However, their *F1-scores* (around 87.8% – 88.7%) were slightly lower than GOA's 92.5%. This suggests that while they achieved similar overall classification *Accuracy*, GOA has a better balance between *Precision* and *Recall* in identifying fake profiles.

In contrast, SVM, K-NN, and K-means displayed lower overall performance compared to GOA, ID3, NB, and RF. Their *Accuracy* ranged from 67% (for K-means) to 0.856 (for K-NN with $k = 3$), and *F1-scores* fell within a similar range. This indicates that these algorithms are not as effective in capturing the complex relationships between features and identifying fake profiles as GOA and the other well-performing algorithms.

Table 8 summarizes the comparison of obtained Accuracy with the original dataset source [21]. Table 8 shows that GOA achieves an Accuracy of 93.9%, which is not just marginally better but significantly higher than its closest competitors: RF (89.2%) and ID3 (89.0%). The absolute differences of 4.7% and 4.9%, respectively, are substantial in ML terms.

- In high-stakes domains like fake profiles detection, this means, for example, that in a dataset of 1 million profiles, GOA would correctly classify approximately 47,000 more profiles than RF and approximately 49,000 more than ID3.

- In terms of error reduction, GOA reduces errors by 43.5% compared to RF and by 44.5% compared to ID3.

GOA uses LR as its Fitness function, which is fundamentally a linear model with a sigmoid activation.

- RF at 89.2% is an ensemble of decision trees, effectively handling complex, non-linear relationships.

- SVM at 84.4% can use kernel tricks to map data into higher-dimensional spaces, improving separation.
- ID3 at 89.0% and RF both capture hierarchical, non-linear decision boundaries.
- GOA, with its LR Fitness function, outperforms these methods, suggesting that the problem of fake profiles detection might involve more linearly separable components than initially expected. More specifically, GOA's optimization capabilities compensate for the simplicity of LR, achieving a near-optimal linear boundary.
- GOA achieving a "near-optimal linear boundary" suggests that it has been successful in finding a linear decision boundary that is very close to the ideal solution. It implies that :
  - The problem of fake profiles detection may be more linearly separable than initially anticipated. This means that there might be a clear and distinct dividing line between real and fake profiles, making it easier for a linear model to classify them (linear separability).
  - GOA's optimization capabilities have been effective in finding this near-optimal linear boundary. This indicates that GOA is well-suited for problems with linearly separable structures.

The proposed approach demonstrated resistance to overfitting. Small datasets are usually prone to overfitting, where models learn noise rather than true patterns.

- GOA's high Accuracy (93.9%) on a small dataset indicates strong regularization properties.
- LR's linear nature inherently resists overfitting compared to high-capacity models.
- GOA's nature-inspired search helps avoid local optima that can lead to overfitting.
- In contrast, RF and ID3, with their ability to create complex trees, are more susceptible to overfitting on small datasets. However, they still perform well (89.2% and 89.0%), indicating the dataset's inherent learnability.

Regarding the dataset characteristics, the high Accuracy across most algorithms (except k-means) suggests that:

- The dataset, though small, has clear, distinguishable features.
- The classes (fake and real profiles) have distinct linguistic or metadata patterns.
- NB uses Bayes' theorem with a strong independence assumption and direct probability estimation but no optimization.
- In contrast, GOA, optimizes a LR model by balancing exploration and exploitation.
- Table 8 shows a cutting-edge algorithm (GOA) surpassing a classical one (NB).

What about remaining algorithms.

- SVM (84.4%) underperforms compared to its usual high standards, which suggests a feature scaling issue or that the optimal hyperplane is more nonlinear than SVM can effectively model.
- K-NN's with k=3 (85.6%) shows that local neighborhoods are somewhat indicative of class, but its inferiority to GOA suggests that global patterns (captured by LR) are more discriminative than local ones.
- K-means' poor 67% performance reveals that unsupervised clustering doesn't align well with the supervised classes. Fake and real profiles might share surface-level similarities that confuse k-means.

A few words about GOA's optimization features:

- GOA models grasshoppers' attraction, repulsion, and wind-guided movement. In the ML context:
  - Attraction: Moves solutions toward promising areas (exploitation).
  - Repulsion: Maintains diversity, avoiding premature convergence (exploration).
  - Wind guidance: Uses the best solution to guide others, similar to gradient information in gradient descent.
- This balanced exploration-exploitation likely helps GOA find a globally optimal LR model, explaining its superior performance.

In fake profiles detection, Precision (not flagging real profiles as fake) is crucial to maintain trust. If GOA's 93.9% Accuracy represents balanced Precision and Recall, it would minimize both types of errors: Fewer real profiles wrongly flagged, maintaining platform credibility, and more fake profiles caught, reducing the spread of computer crime. However, scalability questions remain:

- GOA's computational complexity with millions of features (words, bigrams, etc.).

- Whether its grasshopper model's dynamics scale effectively to higher dimensions.

5. Conclusion

In this study, a novel approach for fake profiles detection on social media platforms is introduced, leveraging GOA with a specific Fitness function. Extensive experimentation and analysis have yielded several significant findings that not only validate the approach's effectiveness but also offer broader insights into the domains of ML and social media security.

Firstly, the proposed approach achieved a remarkable 93.9% Accuracy in fake profiles identification, substantially outperforming a wide array of traditional and modern ML algorithms. This superior performance is not merely a numerical advantage; it translates to tangible real-world benefits. In a hypothetical dataset of one million profiles, this approach would correctly classify approximately 47,000 more profiles than RF and 49,000 more than ID3.

Secondly, this study sheds light on the nature of the fake profiles detection problem itself. Despite its apparent complexity, the high performance of this nature-inspired metaheuristic-based approach suggests that the problem may have more linearly separable components than previously thought. This insight challenges the common assumption that such social media issues are inherently nonlinear.

Moreover, this contribution underscores the potential of nature-inspired metaheuristics in ML. GOA, mimicking the social behavior of grasshoppers, demonstrates a remarkable ability to navigate the solution space effectively. Its mechanisms of attraction, repulsion, and wind-guided movement elegantly translate into a balanced exploration-exploitation strategy, enabling it to find near-optimal solutions even with a simple LR base. This success invites further exploration of nature-inspired algorithms across various ML tasks.

Our study also offers practical insights for ML practitioners. Despite the small size of the exploited dataset, GOA's high Accuracy suggests strong regularization properties, resisting overfitting where more complex models might falter. This resilience is crucial in real-world scenarios where large, cleanly labeled datasets are often a luxury (private, expensive).

However, this work also points to future research directions. While GOA excels on the current dataset, its scalability to high-dimensional spaces—common in text-rich domains like social media—remains to be fully explored. Additionally, testing the proposed approach on larger, more diverse datasets, would further validate its robustness and generalizability.

References

[1] C. Marinoni and E. Ferrara. The evolution of bot detection techniques on social media. *ACM Computing Surveys*, 57(3):1–38, 2024.

[2] Carlo Marinoni, Marco Rizzo, and Maria Assunta Zanetti. Fake profiles and time spent online during the covid 19 pandemic: a real risk for cyberbullying? *Current Psychology*, pages 1–9, 2024.

[3] M Albayati and A Altamimi. Mdfp: a machine learning model for detecting fake facebook profiles using supervised and unsupervised mining techniques. *International Journal of Simulation: Systems, Science & Technology*, 20(1):1–10, 2019.

[4] Julian Aron Prenner and Romain Robbes. Making the most of small software engineering datasets with modern machine learning. *IEEE Transactions on Software Engineering*, 48(12):5050–5067, 2021.

[5] Jungho Kim, Ziqi Wang, and Junho Song. Adaptive active subspace-based metamodeling for high-dimensional reliability analysis. *Structural Safety*, 106:102404, 2024.

[6] Muxuan Han, Zunfeng Du, Kum Fai Yuen, Haitao Zhu, Yancang Li, and Qiuyu Yuan. Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Systems with Applications*, 239:122413, 2024.

[7] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*, 105:30–47, 2017.

[8] W Rose Varuna, K Shalini, and Maria Elna Akkalya Roy. An efficient framework for fake profile identification using metaheuristic and deep learning techniques. *Journal of Positive School Psychology*, pages 3741–3750, 2022.

[9] Nadir Mahammed, Souad Bennabi, Mahmoud Fahsi, Badia Klouche, Nadia Elouali, and Chourouk Bouhadra. Fake profiles identification on social networks with bio inspired algorithm. In *2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA)*, pages 48–52. IEEE, 2022.

[10] Nawel Sekkal, Nadir Mahammed, and Zouaoui Guellil. A bio-inspired grey wolf approach to enhancing fake profile detection in online social media. *Ingénierie des Systèmes d'Information*, 30(4), 2025.

[11] Nadir Mahammed, Badia Klouche, Imène Saidi, Miloud Khaldi, and Mahmoud Fahsi. Bio-inspired algorithms for effective social media profile authenticity verification. In *6th International Hybrid Conference On Informatics And Applied Mathematics*, pages 109–119. CEUR Workshop Proceedings, 2023.

[12] Dharmaraj R Patil, Tareek M Pattewar, Vipul D Punjabi, and Shailendra M Pardeshi. Detecting fake social media profiles using the majority voting approach. *EAI Endorsed Transactions on Scalable Information Systems*, 2024.

[13] Soorya Ramdas and Neenu NT Agnes. Leveraging machine learning for fraudulent social media profile detection. *Cybernetics and Information Technologies*, 24 (1):118–136, 2024.

[14] Nadir Mahammed, Badia Klouche, Imène Saidi, Miloud Khaldi, and Mahmoud Fahsi. Enhancing social media

profile authenticity detection: A bio-inspired algorithm approach. *Machine Learning for Networking*, 14525:32–49, 2024.

[15] Chanchal Kumar, Taran Singh Bharati, and Shiv Prakash. Online social networks: An efficient framework for fake profiles detection using optimizable bagged tree. In *International Conference on Data & Information Sciences*, pages 255–264. Springer, 2024.

[16] Eswara Venkata Sai Raja, Bhrugumalla LVS Aditya, and Sachi Nandan Mohanty. Fake profile detection using logistic regression and gradient descent algorithm on online social networks. *EAI Endorsed Transactions on Scalable Information Systems*, 11(1), 2024.

[17] M Sri Raghavendra, P Penchala Prasad, E Sai Neha, K Meghana, and CH Buela. Utilizing machine learning techniques to eradicate fake profiles. *Bulletin For Technology And History Journal Issn No*, 391:6715, 2024.

[18] Parita Shah, Hiren Patel, and Priya Swaminarayan. Multitask sentiment analysis and topic classification using bert. *ICST Trans. Scalable Inf. Syst*, 11:1–12, 2024.

[19] G Asmat and KM Maiyama. Bitcoin price prediction using n-beats ml technique. *EAI Endorsed Transactions on Scalable Information Systems*, 12(2), 2025.

[20] Salman Farsi and Mahfuzulhoq Chowdhury. Ecom-fraudex: An explainable machine learning framework for victim-centric and dual-sided fraud incident classification in e-commerce. *EAI Endorsed Transactions on Scalable Information Systems*, 12(2), 2025.

[21] Buket Erşahin, Özlem Aktaş, Deniz Kılınç, and Ceyhun Akyol. Twitter fake account detection. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 388–392. IEEE, 2017.

[22] Naga Ramesh Palakurti and Nageswararao Kanchepu. Machine learning mastery: Practical insights for data processing. In *Practical Applications of Data Processing, Algorithms, and Modeling*, pages 16–29. IGI Global, 2024.

[23] Marco Siino, Ilenia Tinnirello, and Marco La Cascia. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342, 2024.

[24] Usman Naseem, Imran Razzak, and Peter W Eklund. A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. *Multimedia Tools and Applications*, 80:35239–35266, 2021.

[25] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01): 20–28, 2021.

[26] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020.

[27] Rizkan Zulyadi, S Surjanti, Ardhana Januar Mahardhani, Sardjana Orba Manullang, Albertus Dwiyoga Widiantoro, and S Suprihatin. K-nearest neighbors method prediction of the anti-corruption behavior index by region of residence. In *AIP Conference Proceedings*, volume 3001. AIP Publishing, 2024.

[28] M Afriansyah, Joni Saputra, Valian Yoga Pudya Ardhana, and Yuan Sa'adati. Algoritma naive bayes yang efisien untuk klasifikasi buah pisang raja berdasarkan fitur warna. *Journal of Information Systems Management and Digital Business*, 1(2):236–248, 2024.

[29] Zhigang Sun, Guotao Wang, Pengfei Li, Hui Wang, Min Zhang, and Xiaowen Liang. An improved random forest based on the classification accuracy and correlation measurement of decision trees. *Expert Systems with Applications*, 237:121549, 2024.

[30] SS Kavitha and Narasimha Kaulgud. Quantum machine learning for support vector machine classification. *Evolutionary Intelligence*, 17(2):819–828, 2024.

[31] Meetu Jain, Vibha Saihjpal, Narinder Singh, and Satya Bir Singh. An overview of variants and advancements of pso algorithm. *Applied Sciences*, 12(17): 8392, 2022.

[32] Yassine Meraihi, Asma Benmessaoud Gabis, Seyedali Mirjalili, and Amar Ramdane-Cherif. Grasshopper optimization algorithm: theory, variants, and applications. *Ieee Access*, 9:50001–50024, 2021.

[33] Shruti Shinde and Sunil B Mane. Hybrid approach for fake profile identification on social media. In *Pattern Recognition and Data Analysis with Applications*, pages 579–590. Springer, 2022.