The Eurasia Proceedings of Science, Technology, Engineering and Mathematics (EPSTEM), 2025

Volume 38, Pages 849-860

IConTES 2025: International Conference on Technology, Engineering and Science

A Bio-Inspired Path to Fake Profile Detection: Revisiting Linear Models Through Grasshopper Optimization

Nadir Mahammed
Djillali Liabes University of Sidi Bel Abbes

Imene Saidi
Djillali Liabes University of Sidi Bel Abbes

Mahmoud Fahsi
Djillali Liabes University of Sidi Bel Abbes

Souad Bennabi
Hassiba Ben Bouali University of Chlef

Abstract: Social media platforms continue to struggle with the proliferation of fake user profiles, which undermine trust and facilitate the spread of misinformation. While deep learning and complex ensemble models dominate recent solutions, this study revisits the power of simpler classifiers when paired with intelligent feature selection. We introduce a hybrid framework that couples the Grasshopper Optimization Algorithm with Logistic Regression for detecting fake profiles in online social networks. The proposed bio-inspired algorithm mimics the collective foraging behavior of grasshoppers to optimize the feature space, resulting in a compact and highly discriminative set of inputs. Evaluated on different datasets from social media platforms, the proposed model not only outperforms six standard classifiers but also challenges the assumption that fake profile detection requires non-linear modeling. Achieving impressive accuracy and strong F1 performance, our approach shows that nature-inspired metaheuristics can elevate classical models to competitive levels. These results suggest a promising direction for lightweight, interpretable, and scalable solutions in social cybersecurity.

Keywords: Fake accounts, Grasshopper optimization algorithm, Feature selection, Logistic regression, Online social networks

Introduction

In the digital age, social media platforms have become modern town squares, where people connect and share information like never before. But this openness has a downside: the widespread use of fake profiles (Fire et al., 2014). These fake accounts—often run by bots or malicious actors—pose a real threat to online conversations. They're used to spread misinformation, influence public opinion, and enable cybercrime. According to Marinoni et al. (2024), up to 15% of social media accounts could be bots, showing how serious the problem is.

Catching these fake profiles isn't simple. Many of them are designed to act like real users, using advanced language tools and sometimes even stealing real account details (Albayati & Altamimi, 2019). Old detection systems based on fixed rules often fail because attackers constantly change tactics. Machine learning offers a more flexible solution by learning patterns that rule-based systems miss. But even machine learning has limits. Training these models requires a lot of labeled data, which is slow and expensive to produce (Prenner & Robbes, 2021). Also, the data involved—like user behavior, connections, and text—is complex (Kim et al., 2024), and the boundary between real and fake profiles is rarely clear-cut.

- Selection and peer-review under responsibility of the Organizing Committee of the Conference

Newer approaches use algorithms inspired by nature to tackle this kind of complexity (Han et al., 2024). One example is the Grasshopper Optimization Algorithm (GOA), which models how grasshoppers move in groups, balancing attraction, repulsion, and wind-driven movement (Saremi et al., 2017). GOA has done well in other optimization tasks, but its use in machine learning—especially for detecting fake profiles—hasn't been studied much.

This paper tries to close that gap. The authors suggest a new way to detect fake accounts by combining GOA with a specific fitness function. Their main assumptions are:

- GOA can explore the structure of social media data more effectively than traditional optimizers, even when used with a basic machine learning model.
- Despite how complex the problem seems, fake profile detection might involve patterns that are easier to separate than previously thought.

The paper is organized into six sections. Section 2 reviews current methods for detecting fake profiles. Section 3 explains the data, preprocessing, and the proposed approach. Section 4 presents the results and discusses them. Section 5 concludes the study.

Related Work

This section gives a quick look at different ways researchers have tried to detect fake profiles on social media using machine learning.

Varuna et al. (2022) introduced a system for spotting fake profiles that uses open-source big data tools along with a type of neural network called LSTM (Long Short-Term Memory). They also applied a method called Dispersive Flies Optimization (DFO) to pick out the most useful features from the data. Their approach focused on collecting data ethically and looked at both public and private user information. Mahammed et al. (2022) focused on fake accounts on Facebook. They came up with a two-step system that first uses the Satin Bowerbird Optimization Algorithm (SBO) to group profiles and find the best starting points, then apply K-Means clustering to classify each profile as real or fake. In a later study, Sekkal and Mahammed (2025) compared different fake profile detection methods. They found that both supervised and unsupervised machine learning models show promise, especially those inspired by nature. They tested an algorithm based on the behavior of grey wolves—called the Grey Wolf Optimizer (GWO)—to detect fake accounts. Another study by Mahammed et al. (2025) used a nature-inspired method called the Fire Hawk Optimizer (FHO). They tested different groups of features from a Twitter dataset to see which ones were most effective. They used Gradient Boosting Classifier (GBC) as their main model to evaluate how well FHO worked. Patil et al. (2024) proposed a different approach. They combined several machine learning models into one system and used a Majority Voting Technique (MVT) to decide if a profile was fake or real. Their results showed strong potential for improving online safety. Ramdas and Agnes (2024) also explored machine learning for detecting fake profiles. They tested various algorithms and used evaluation tools like confusion matrices and error rate analysis to find the most effective models. In a more recent paper, Mahammed et al. (2024) again investigated the Fire Hawk Optimizer, this time using it with Facebook data. They used Gradient Boosting Classifier to measure its performance and focused on finding the most useful features to tell real and fake accounts apart. Kumar et al. (2024) took a different route. They used a large set of features—including profile details, user behavior, and network connections—and applied a refined version of the Bagged Tree Algorithm (BTA). This method prunes unnecessary parts of the decision tree to improve both accuracy and speed.

Table 1 summarizes all these studies and shows how researchers are making progress in detecting fake profiles across platforms like Facebook, Twitter, and Instagram. Most studies use machine learning and report very high accuracy—often over 97%. Three of them (Mahammed et al., 2022, 2023, 2024) focus on combining machine learning with metaheuristic algorithms like SBO and FHO. These methods also perform well, with accuracy ranging from 98.0% to 99.8%, but more research is needed to confirm how well they generalize. Some of the most commonly used tools include K-Means, Gradient Boosting Classifier, Random Forest, and other ensemble techniques. One study (Varuna et al., 2022) stands out for using LSTM networks, which are particularly good at analyzing content over time, like sequences of text. It's worth noting that while many studies combine machine learning with optimization techniques for better results, the effectiveness of any method may depend on the platform and the type of user data available. Dataset sizes vary from just over 1,200 to more than 17,000 entries. While larger datasets generally lead to better results, some smaller studies still achieved high accuracy, suggesting that choosing the right method matters as much as the amount of data. So far, machine learning has been very successful at spotting fake accounts. But algorithms inspired by nature—like those based on bird, insect, or animal behavior—are still not widely used. This work proposes a new approach that brings those two ideas together: combining machine learning with a nature-inspired algorithm to improve fake profile detection on social media.

Table 1. Related work summary

Reference	OSN	ML	Metaheuristic	Other	Dataset size	Results (acc)
Varuna et al. (2022)	Facebook	LSTM	DFO	-	-	0.979
Mahammed et al. (2022)	Facebook	k-means	SBO	hybridization	1244	0.989
Sekkal and Mahammed (2025)	Facebook	-	GWO	Transition	1244	0.980
Mahammed et al. (2025)	Twitter	GBC	FHO	Hybridization	17350	0.996
Patil et al. (2024)	Twitter	MVT	-	Combination	6825	0.991
Ramdas and Agnes (2024)	Instagram	RF	-	Comparison	6868	0.997
Mahammed et al. (2024)	Facebook	GBC	FHO	Hybridization	1244	0.998
Kumar et al. (2024)	Facebook	BTA	-	-	-	0.999

Material and Methodology

This section explains the steps taken in the study, starting with the dataset used. It then summarizes how the data was prepared for analysis, followed by an overview of the machine learning methods considered. Finally, it introduces the main technique used in the study—a bio-inspired algorithm—with a brief explanation of how it works, why it was combined with other methods, and how it functions.

Dataset

The study used a balanced dataset from Twitter, collected by Erşahin et al. (2017). It contains 1,000 user profiles, split into 501 fake and 499 real accounts. This even distribution helps ensure reliable and unbiased analysis. The dataset includes 16 features, listed in Table 3. The breakdown of real and fake profiles is shown in Table 2.

Table 2. Twitter dataset repartition

	Real	Fake
Record	499	501
Proportion (%)	49.9	50.1

Dataset Preprocessing

Before any analysis can begin, the raw data needs to be cleaned up and organized. This step—called preprocessing—is essential because the original data often contains problems like missing information, inconsistent formatting, or extra noise that can throw off results (Siino et al., 2024). Especially in machine learning, the data has to be neat and structured for the algorithms to work well. Taking care of these issues early on helps ensure the analysis is accurate and reliable (Naseem et al., 2021).

Figure 1 shows the 12-step process used in this study to clean and prepare the Twitter data for analysis (Naseem et al., 2021). Here's a breakdown of what those steps involved:

- Step 1: Clean up the text by removing things like special characters, links, user tags, and hashtags so the focus stays on the actual content.
- Steps 2 & 3: Add meaning by translating emojis and emoticons into words that express their emotions, and expanding abbreviations and acronyms to their full forms for better clarity.
- Step 4: Fix spelling errors to make the text more accurate and easier to analyze.
- Steps 5 & 6: Expand contractions (like “don’t” to “do not”) and shorten stretched-out words (like “soooo” to “so”) to make the text more standard.
- Steps 7 to 11:

– Remove punctuation to simplify the text
– Convert everything to lowercase for consistency
– Separate words clearly for better analysis
– Remove numbers if they’re not useful for the task
– Get rid of common words like “the” or “is” that don’t add much meaning

- Step 12: Reduce words to their basic form (for example, changing “running” to “run”) to keep things consistent and easier to analyze.

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 that the user has an account with ”contributor mode” enabled.
default profile	When true, indicates that the user has not altered the theme or background of their user profile.
default profile image	When true, indicates that the user has not uploaded their own profile image and a default image is used instead.
is translator	When true, indicates that the user is a participant in Twitter’s translator community.
hashtags average	Number of hashtags that user has used in last 20 tweets.
mentions average	Number of mentions that user has used in last 20 tweets.
urls average	Number of URL links that user has used in last 20 tweets.

Machine Learning Algorithms

With the Twitter data cleaned and ready, this section gives a quick overview of the machine learning methods used in the study.

- ID3 (Induction of Decision Tree) This is a supervised learning method that builds a decision tree based on patterns found in the training data. The tree is then used to classify new, unseen data (Charbuty and Abdulazeez, 2021).
- K-means is an unsupervised learning method used to group similar data points into clusters. It works by organizing the data into a set number of groups based on similarity, even without labeled data (Sinaga and Yang, 2020).
- K-Nearest Neighbors (KNN) is a supervised method that classifies data based on its closest neighbors. A new data point is labeled according to the majority label among its nearest examples in the training set (Zulyadi et al., 2024).
- Naive Bayes (NB) is a supervised method based on Bayes’ theorem. It assumes all features are independent from each other, which makes it simple and fast while still often producing accurate results (Afriansyah et al., 2024).

- Random Forest (RF) is an ensemble learning method that combines many decision trees. Each tree makes a prediction, and the final decision is based on the majority vote, improving overall accuracy (Sun et al., 2024).
- Support Vector Machine (SVM) is a supervised learning method that finds the best boundary to separate data into categories. Once this boundary is set, it can be used to classify new data with high reliability (Kavitha and Kaulgud, 2024).

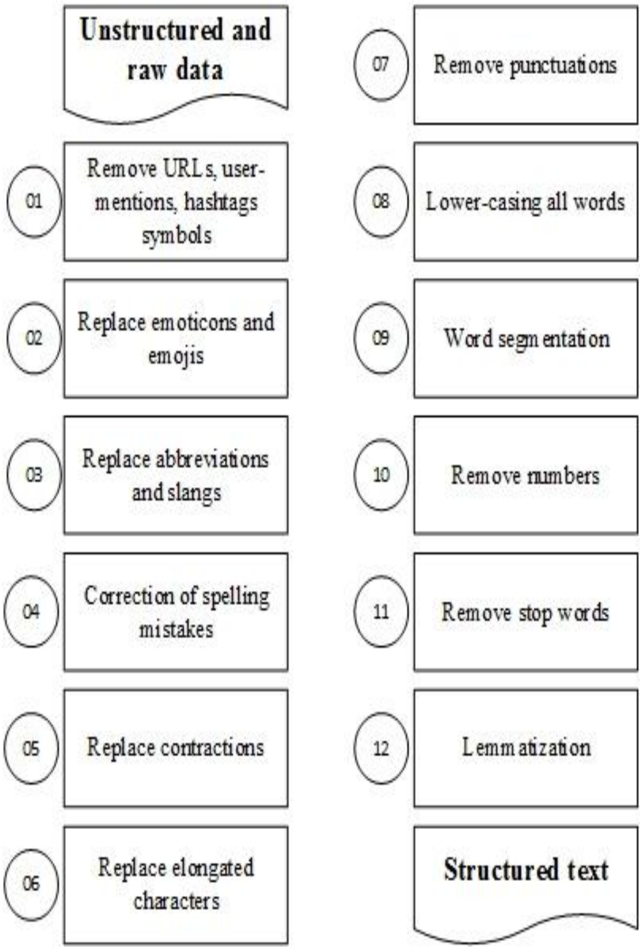

Unstructured and raw data
01
Remove URLs, user-mentions, hashtags symbols
02
Replace emoticons and emojis
03
Replace abbreviations and slangs
04
Correction of spelling mistakes
05
Replace contractions
06
Replace elongated characters
07
Remove punctuations
08
Lower-casing all words
09
Word segmentation
10
Remove numbers
11
Remove stop words
12
Lemmatization
Structured text

Figure 1. Preprocessing pipeline

Proposed Algorithm

This section explains the Grasshopper Optimization Algorithm (GOA) introduced by Saremi et al. (2017). It covers how the algorithm works, the role of the fitness function in measuring its performance, and why GOA is a good fit for this study. The explanation also connects the way the algorithm is designed to the natural behavior it's modeled after, showing how this link supports its use in solving the problem.

Grasshopper Optimization Algorithm

GOA is a type of algorithm that takes inspiration from nature—specifically, how grasshoppers behave in swarms. In this method, each possible solution is treated like a grasshopper exploring a search area. These "grasshoppers" move through the space based on how well they perform, aiming to follow the best-performing one, called the leader (Saremi et al., 2017). This leader keeps changing depending on which solution is currently the best, encouraging the others to move toward it. This mirrors how real grasshoppers swarm toward food sources. What makes GOA different from many other nature-inspired algorithms is its built-in foraging behavior. This means it doesn't just chase the best solution but also explores the search space more thoroughly, which increases its chances of finding better results (Afriansyah et al., 2024).

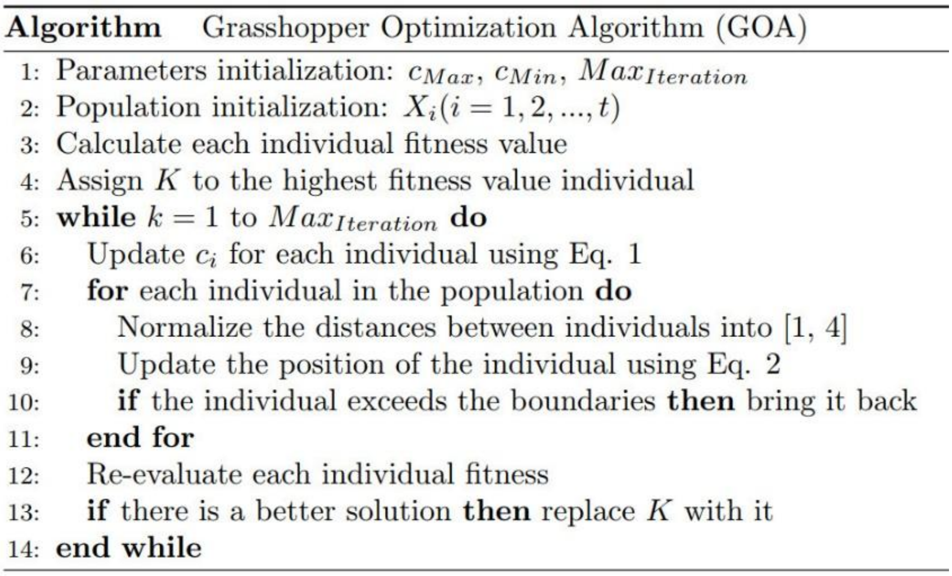
Progress and Functioning

Figure 2 shows the step-by-step outline (pseudo code) of how GOA works.

$$c_i = c_{Max} - \frac{k-1}{Max_{iteration}-1}\left(c_{Max} - c_{Min}\right) \tag{1}$$

The parameters c_{Max} and c_{Min} correspond to the upper and lower bounds for the coefficient c, respectively. Within the context of the algorithm, t denotes the current iteration, progressing from 1 to a maximum value represented by k_{Max}.

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



```
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. Grasshopper optimization algorithm pseudo code

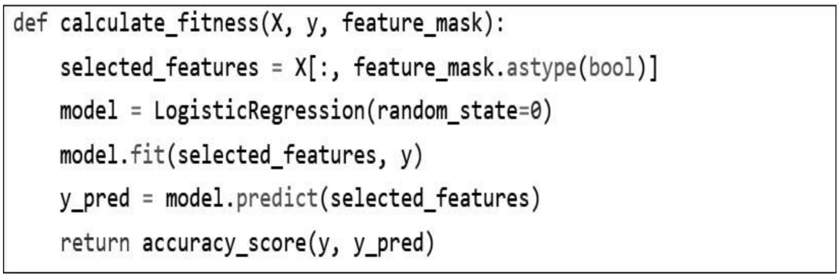
Within each dimension of the search space, the algorithm sets a lower and upper limit, known as lb_d and ub_d. It also keeps track of the best solution found so far in that specific dimension, referred to as $\widehat{K}_d$.

The parameter c_1 works similarly to the inertia weight used in Particle Swarm Optimization (Jain et al., 2022). Its role is to gradually reduce how much the grasshoppers move around the target. This helps the algorithm balance two goals: focusing more on good solutions already found (exploitation) while still exploring new possibilities (exploration). The parameter c_2 is used to slowly shrink the areas where grasshoppers either attract or repel each other. As the algorithm runs through more cycles, these zones get smaller. Often, c_1 and c_2 are combined into a single formula, as shown in equation (1) (Meraihi et al., 2021).

Fitness Function

GOA uses a fitness function to measure how good each solution is. In this case, it uses logistic regression to select the most useful features from the data.

- Feature selection function: As shown in figure 3, this function takes the dataset, the target labels, and a feature mask, which shows which features are selected. It runs a logistic regression model to check how well the selected features can classify the data and returns the accuracy.
- GOA function: As shown in figure 4, this part of the process runs the main GOA logic. It uses inputs like the dataset, target labels, number of grasshoppers, and the number of iterations to run.



```
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 Inference, instead of using a separate, pre-set formula to measure how good a solution is, this approach calculates the fitness value directly inside the grasshopper optimization process. It works by checking how accurate a logistic regression model is when trained on the features chosen by the current "grasshopper," or candidate solution.

To turn this into a minimization problem (where lower scores are better), the accuracy is flipped using the formula $^{1}/_{(1 + accuracy)}$. This inverted score becomes the fitness value for that grasshopper. In short, each candidate solution is judged by how well its selected features help a logistic regression model classify the data—using the flipped accuracy as the measure. To test how well GOA performs, it's compared with several other well-known machine learning methods: ID3, SVM, Naive Bayes, Random Forest, K-Nearest Neighbors (with K=3), and K-means. Each method is tested 100 times. The GOA settings used for these tests are listed in Table 4.

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	-

Results and Discussion

After explaining the method and how it works, this section covers how the model was evaluated, what the results were, and why they matter.

Evaluation Criteria

To measure how well the model performed, the study used a confusion matrix. This is basically a table that compares what the model predicted with what the actual answers were (Shinde & Mane, 2022). The main metric used was accuracy, but other measures like precision, recall, and F1-score are also useful for getting a full picture. The matrix includes the following terms:

* TP: True Positive (correctly identified fake profiles)
* FP: False Positive (real profiles wrongly flagged as fake)
* TN: True Negative (correctly identified real profiles)
* FN: False Negative (fake profiles wrongly flagged as real)

Results and Discussion

The results show that the proposed method is highly effective at spotting fake profiles, as shown in the confusion matrix in Table 5.

- First, the method reached a high accuracy of 93.9%. It correctly identified 432 fake profiles (True Positives) and 496 real profiles (True Negatives), showing that it can reliably tell the difference between real and fake users.
- Second, it's especially good at catching fake accounts. The high number of True Positives means it rarely misses them.
- Third, it makes few mistakes when it comes to real users. With only 29 False Negatives, the method keeps the number of wrongly flagged real accounts low, helping to avoid blocking genuine users by mistake.
- Finally, the results show that the model is well-balanced. It performs consistently across both classes—real and fake profiles—without leaning too heavily in one direction. This balance is key to building a fair and trustworthy system.

Table 5. Confusion matrix

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

Table 6, titled "Obtained Results," shows how different algorithms perform when used to detect fake social media profiles. It includes key performance measures like accuracy, precision, recall, and F1-score. The standout performer is the Grasshopper Optimization Algorithm (GOA). It achieved the highest accuracy at 93.9% and the best F1-score at 92.5%, making it the most effective method among those tested. These results show that GOA does a great job of exploring different possibilities and fine-tuning its approach to deliver strong results. One reason GOA performs so well is its ability to balance two important tasks: exploration (searching widely for possible solutions) and exploitation (focusing on refining the best ones found so far). This balance helps the algorithm avoid getting stuck in weak areas and instead find stronger solutions. GOA is also well-designed for solving complicated problems like fake profile detection, which can involve messy or unpredictable data. Its flexible approach helps it handle patterns and irregularities that might throw off simpler methods. The use of logistic regression in the GOA setup also adds to its strength. Logistic regression is especially good at handling yes-or-no problems—like deciding whether a profile is real or fake—by finding relationships between the data and the outcome.

Other algorithms like ID3, Naive Bayes (NB), and Random Forest (RF) also performed well, with accuracy scores close to GOA's, ranging from 88.9% to 89.2%. However, their F1-scores were slightly lower (around 87.8% to 88.7%), suggesting they weren't quite as balanced in terms of precision and recall. Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and K-means didn't perform as well. KNN reached an accuracy of 85.6%, while K-means was the lowest at 67%. Their F1-scores followed a similar pattern, showing that these models struggled more with capturing the complex patterns needed to reliably identify fake accounts. In short, GOA proved to be the most effective option overall, especially when paired with logistic regression, and stood out by offering both high accuracy and strong balance across all key metrics.

Table 6. 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
KNN (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

Table 7 compares the accuracy of different algorithms using the same dataset originally provided by Erşahin et al. (2017). GOA comes out on top with an accuracy of 93.9%, which is a significant step above the next best performers—Random Forest at 89.2% and ID3 at 89.0%. That's a difference of about 4.7% to 4.9%, which is a big deal in machine learning.

To put this into perspective, if you were working with a dataset of 1 million profiles, GOA would correctly classify around 47,000 more profiles than Random Forest, and about 49,000 more than ID3. GOA also does a better job at reducing errors. Compared to Random Forest, it cuts errors by 43.5%, and compared to ID3, by 44.5%. What's interesting is that GOA uses logistic regression (LR) to measure how good each solution is. Logistic regression is a simple model that draws straight lines to separate classes. In contrast:

- Random Forest builds complex decision trees to capture more complicated patterns.
- SVM uses mathematical tricks to separate data in more complex ways.
- ID3 and Random Forest both look for layered, non-linear boundaries in the data.

Despite being simpler, GOA's use of logistic regression actually performs better. This suggests that the fake profile problem may not be as complex as expected and that a good linear solution—found through optimization—might be enough.

Another strong point is that GOA seems to avoid overfitting, which is when a model memorizes the training data instead of learning real patterns. This is impressive given the small dataset. Logistic regression naturally avoids overfitting because of its simplicity, and GOA's swarm-based search avoids getting stuck in bad solutions. On the other hand, Random Forest and ID3, which can overfit on small datasets due to their complexity, still did well—showing the dataset is also quite learnable.

Table 7. Accuracy comparison

Algorithm	Ersahin et al. (2017)	Our propsotion
ID3	-	0.890
SVM	-	0.844
KNN (k=3)	-	0.856
NB	0.909	0.889
RF	-	0.892
K-means	-	0.67
GOA	-	0.939

Looking at the dataset itself, the high accuracy across most algorithms (except k-means) suggests the features are pretty clear. There seem to be strong differences in how fake and real profiles are written or structured. Naive Bayes, which assumes all features are independent, works without optimization. GOA, however, uses a modern optimization approach inspired by how grasshoppers move in swarms and fine-tunes a logistic regression model by balancing broad searching and focused improvements. The results show GOA outperforms older models like Naive Bayes. As for the other algorithms:

- SVM performed below its usual level (84.4%), which could mean the features weren't scaled properly or the decision boundary was too complex.
- KNN (with k=3) scored 85.6%, showing that nearby data points share similar traits, but it didn't do as well as GOA, suggesting that broader patterns matter more.
- K-means had the lowest performance at 67%, likely because it's unsupervised and doesn't use label information. Fake and real profiles can look similar on the surface, which makes clustering harder.

GOA's design helps explain its success. It mimics how grasshoppers move in nature:

- Attraction pulls solutions toward better areas.
- Repulsion keeps the search diverse and avoids getting stuck.
- Wind guidance helps all solutions move in the direction of the best one found so far.

This balance between exploring and improving is probably what helps GOA find such an effective logistic regression model. In fake profile detection, precision really matters. You want to avoid mistakenly flagging real people as fake, because that can damage trust. If GOA's 93.9% accuracy includes a strong balance between precision and recall, it means it's doing both things well—catching fake accounts while minimizing harm to real users. That said, there are still open questions about how well GOA would scale. Its optimization process might become more demanding if it's applied to much larger datasets with millions of features, like words or bigrams. It's also unclear whether the grasshopper-inspired behavior would hold up in those more complex spaces.

Conclusion

This study introduced a new way to detect fake profiles on social media using the Grasshopper Optimization Algorithm (GOA) combined with a specific fitness function. The method was tested thoroughly, and the results were both impressive and meaningful—not just for improving detection accuracy, but also for what they reveal about how these problems can be tackled.

The approach reached an accuracy of 93.9%, beating several well-known machine learning models. That's not just a slight edge—it means that, in a dataset of one million profiles, this method would correctly identify around 47,000 more fake accounts than Random Forest and about 49,000 more than ID3. In a world where fake profiles can spread misinformation, commit fraud, or manipulate public opinion, this improvement could make a real difference in protecting online communities and building trust.

Interestingly, while detecting fake profiles might seem like a complex problem, the success of this method—which is based on a simple logistic regression model optimized by a nature-inspired algorithm—suggests it might be more straightforward than we thought. This challenges the idea that social media problems always require highly complex solutions and opens the door to using simpler, more efficient models in future research.

The study also highlights how powerful nature-inspired algorithms can be. GOA is based on how grasshoppers move in swarms, using behaviors like attraction, repulsion, and movement guided by the strongest member of the group. These behaviors help the algorithm explore different solutions and zero in on the best ones. Even when paired with a basic model like logistic regression, GOA was able to find solutions that outperformed more complex methods. For machine learning practitioners, the results offer practical lessons too. GOA performed well on a relatively small dataset, showing that it can avoid overfitting—a common problem when models are too complex for the amount of data available. This makes it especially useful in real-world cases where clean, large-scale data isn't always available. At the same time, there are areas for future work. We still need to see how well GOA handles much larger and more complex datasets, especially in text-heavy environments like social media. Testing it across more diverse scenarios will help confirm how widely it can be applied. In a time when digital manipulation and online deception are real threats to information and trust, even small improvements in detecting fake profiles can have a big impact. Better detection tools can help limit the spread of false information and create safer, more reliable spaces for people to connect online. This study aims to contribute to that goal.

Scientific Ethics Declaration

* The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the authors.

Conflict of Interest

* The authors declare that they have no conflicts of interest

Funding

* This study was funded by the ministry of research and higher education of Algeria with the code C00l07es220120220004 under the university research and training project. The authors express their gratitude for the support provided.

Acknowledgements or Notes

* This article was presented as an oral presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Türkiye on November 12-15, 2025.

References

Afriansyah, M., Saputra, J., Ardhana, V. Y. P., & Saadati, Y. (2024). 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.

Albayati, M., & Altamimi, A. (2019). 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.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends, 2*(1), 20–28.

Ersahin, B., Aktaş, Ö., Kılınç, D., & Akyol, C. (2017). Twitter fake account detection. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 388–392). IEEE.

Fire, M., Kagan, D., Elyashar, A., & Elovici, Y. (2014). Friend or foe? Fake profile identification in online social networks. *Social Network Analysis and Mining, 4*, 1–23.

Han, M., Du, Z., Yuen, K. F., Zhu, H., Li, Y., & Yuan, Q. (2024). Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Systems with Applications, 239*, 122413.

Jain, M., Saihjpal, V., Singh, N., & Singh, S. B. (2022). An overview of variants and advancements of PSO algorithm. *Applied Sciences, 12*(17), 8392.

Kavitha, S., & Kaulgud, N. (2024). Quantum machine learning for support vector machine classification. *Evolutionary Intelligence, 17*(2), 819–828.

Kim, J., Wang, Z., & Song, J. (2024). Adaptive active subspace-based metamodeling for high-dimensional reliability analysis. *Structural Safety, 106*, 102404.

Kumar, C., Bharati, T. S., & Prakash, S. (2024). Online social networks: An efficient framework for fake profiles detection using optimizable bagged tree. In *International Conference on Data & Information Sciences* (pp. 255–264). Springer.

Mahammed, N., Bennabi, S., Fahsi, M., Klouche, B., Elouali, N., & Bouhadra, C. (2022). Fake profiles identification on social networks with bio-inspired algorithm. In *First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA 2022)* (pp. 48–52). IEEE.

Mahammed, N., Saidi, I., Bencherif, K., Khaldi, M., Fahsi, M., & Guellil, Z. (2025). Grasshopper-based detection of fake social media profiles. *EAI Endorsed Transactions on Scalable Information Systems, 12*(4).

Mahammed, N., Saidi, I., Khaldi, M., & Fahsi, M. (2024). Enhancing social media profile authenticity detection: A bio-inspired algorithm approach. In *Machine Learning for Networking* (Vol. 14525, pp. 32–49). Springer.

Marinoni, C., Rizzo, M., & Zanetti, M. A. (2024). Fake profiles and time spent online during the COVID-19 pandemic: A real risk for cyberbullying? *Current Psychology*, 1–9.

Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper optimization algorithm: Theory, variants, and applications. *IEEE Access, 9*, 50001–50024.

Naseem, U., Razzak, I., & Eklund, P. W. (2021). A survey of preprocessing techniques to improve short-text quality: A case study on hate speech detection on Twitter. *Multimedia Tools and Applications, 80*, 35239–35266.

Patil, D. R., Pattewar, T. M., Punjabi, V. D., & Pardeshi, S. M. (2024). Detecting fake social media profiles using the majority voting approach. *EAI Endorsed Transactions on Scalable Information Systems*.

Prenner, J. A., & Robbes, R. (2021). Making the most of small software engineering datasets with modern machine learning. *IEEE Transactions on Software Engineering, 48*(12), 5050–5067.

Ramdas, S., & Agnes, N. N. (2024). Leveraging machine learning for fraudulent social media profile detection. *Cybernetics and Information Technologies, 24*(1), 118–136.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47.

Sekkal, N., Mahammed, N., & Guellil, Z. (2025). A bio-inspired grey wolf approach to enhancing fake profile detection in online social media. *Ingénierie des Systèmes d'Information, 30*(4).

Shinde, S., & Mane, S. B. (2022). Hybrid approach for fake profile identification on social media. In *Pattern Recognition and Data Analysis with Applications* (pp. 579–590). Springer.

Siino, M., Tinnirello, I., & La Cascia, M. (2024). 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.

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access, 8*, 80716–80727.

Sun, Z., Wang, G., Li, P., Wang, H., Zhang, M., & Liang, X. (2024). An improved random forest based on the classification accuracy and correlation measurement of decision trees. *Expert Systems with Applications, 237*, 121549.

Varuna, W. R., Shalini, K., & Roy, M. E. A. (2022). An efficient framework for fake profile identification using metaheuristic and deep learning techniques. *Journal of Positive School Psychology*, 3741–3750.

Zulyadi, R., Surjanti, S., Mahardhani, A. J., Manullang, S. O., Widiantoro, A. D., & Suprihatin, S. (2024). K-nearest neighbors method prediction of the anti-corruption behavior index by region of residence. In *AIP Conference Proceedings* (Vol. 3001). AIP Publishing.

Author(s) Information

Nadir Mahammed
Djillali Liabes University of Sidi Bel Abbes
LabRI-SBA, Ecole Superieure en Informatique Sidi Bel Abbes, Algeria, P.O. 73, El Wiam, 22016 Sidi Bel Abbès, Algeria
Contact e-mail: *n.mahammed@esi-sba.dz*

Imene Saidi
Djillali Liabes University of Sidi Bel Abbes
LabRI-SBA, Ecole Superieure en Informatique Sidi Bel Abbes, Algeria, P.O. 73, El Wiam, 22016 Sidi Bel Abbès, Algeria

Mahmoud Fahsi
Djillali Liabes University of Sidi Bel Abbes, EEDIS,Algeria
P.O. 89, Sidi Bel Abbès, 22000, Algeria

Souad Bennabi
Hassiba Ben Bouali University of Chlef, LCMCR, Algeria
Hay Salem, n°19, Chlef, 02000, Algeria

To cite this article:

Mahammed, N., Saidi, I., Fahsi, M., & Bennabi, S. (2025). A bio-inspired path to fake profile detection: revisiting linear models through Grasshopper optimization. *The Eurasia Proceedings of Science, Technology, Engineering and Mathematics (EPSTEM), 38,* 849-860.