Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education And Scientific Research

University of Tlemcen – ABU BAKR BELKAID - Algeria
FACULTY OF SCIENCE
DEPARTMENT of COMPUTER SCIENE

Graduation Project

To obtain a Master's degree in Computer Science

Specialty: Intelligent Model and Decision (M.I.D)

On the subject:

Conception and the Implementation of a system for the detection of fake profiles On online social network using machine learning and the bio-inspired algorithms

---

Presented by :

M^{lle} HADJOU SEMIR Fatima zohra khawla

Soutenu le : 30/09/2023

Before the jury composed of:

Chairman :	M^{r} BERRABAH Sid Ahmed	Maître de conférences A
Examiner:	M^{r} BRIKCI NIGASSA Amine	Maître de Assistant A
Supervisor:	M^{me} ILES Nawal	Maître de conférences B
Co-Supervisor:	M^{r} Mahammed Nadir	Maître de conférences A

College year 2022/2023

Acknowledgements

I would like to begin by expressing my deepest gratitude to Almighty Allah for granting me the strength and ability to successfully complete this work. First and foremost, I am immensely thankful to Mr. MAHAMMED Nadir, our teacher and co-supervisor, for his unwavering guidance and tremendous support throughout the entire preparation phase of this dissertation. His constant follow-up and encouragement played a pivotal role in shaping this work. I hold to also thank my supervisor, Mme . ILES Nawal, for her dedicated time, interest, and understanding. Her valuable insights and feedback greatly contributed to the overall quality of this research. Furthermore, I would like to express my sincere thanks to the esteemed members of the jury for graciously agreeing to examine, evaluate, and judge my work. Your expertise and feedback have been invaluable in refining and validating my research. I cannot let this opportunity pass without acknowledging the unwavering support of my GRANDPARENTS and entire family. Their prayers, encouragement, and unwavering belief in me have been my source of strength throughout this journey. I am forever grateful for their love and unwavering support. Last but not least, I would like to extend my gratitude to all those who directly or indirectly contributed to the execution of this work. Your assistance, whether through valuable discussions, technical support, or moral encouragement, has played an integral role in the completion of this dissertation.

I will not let this opportunity pass, without thanking my parents and all my family who , through their prayers and encouragement, i was able to overcome all the obstacles. I would like to thank everyone who participated directly or indirectly in the execution of this work.

Dedication

I dedicate this modest work to those closest to my heart: To my grandparents and my mother who accompanied me during the toughest moments of this long journey of my education , who shared with me all the emotional moments and who always supported and encouraged me . I wish to show my great gratification which will never be enough and that I hope to make them proud of this work.

To my father God bless his soul and eternal peace. To my dear sisters and brothers, this dedication is a testament to the bond we share as a family. I wish you all happiness, good health, and continued success in all your endeavors. Lastly, I extend this dedication to all the members of BENAYED and HADJOU SEMIR family , friends especially TABET AOUL chaimaa, whose love, encouragement, and support have been instrumental in my achievements. I am grateful for your presence in my life. May this dedication serve as a small token of my gratitude and love for all those who have been a part of my journey.

Résumé

La prolifération des activités en ligne sur les réseaux sociaux en ligne (OSN) a attiré une attention considérable des utilisateurs. Cependant, cette croissance a été entravée par l'émergence de comptes frauduleux qui ne représentent pas de véritables individus et violent les réglementations en matière de confidentialité au sein des communautés de réseaux sociaux. Par conséquent, il est impératif d'identifier et de supprimer ces profils pour renforcer la sécurité des utilisateurs OSN. Ces dernières années, les chercheurs se sont tournés vers l'apprentissage automatique (ML) pour développer des stratégies et des méthodes permettant de résoudre ce problème. De nombreuses études ont été menées dans ce domaine pour comparer diverses techniques basées sur le ML. Cependant, la littérature existante manque encore d'un examen complet, en particulier en ce qui concerne les différentes plateformes OSN. De plus, l'utilisation d'algorithmes bio-inspirés a été largement négligée.

Notre étude adopte une approche nouvelle en effectuant une analyse comparative approfondie de diverses techniques de détection de faux profils sur les réseaux sociaux en ligne. Les résultats de notre étude indiquent que les modèles supervisés, ainsi que d'autres techniques d'apprentissage automatique, ainsi que les modèles non supervisés, sont efficaces pour détecter les faux profils sur les réseaux sociaux. Pour obtenir des résultats optimaux, nous avons incorporé six algorithmes bio-inspirés pour améliorer les performances d'identification des faux profils.résultats .

Mots clé : Réseau social en ligne, faux profil, détection, apprentissage automatique

Abstract

The proliferation of online activities on Online Social Networks (OSNs) has captured significant user attention. However, this growth has been hindered by the emergence of fraudulent accounts that do not represent real individuals and violate privacy regulations within social network communities. Consequently, it is imperative to identify and remove these profiles to enhance the security of OSN users. In recent years, researchers have turned to machine learning (ML) to develop strategies and methods to tackle this issue. Numerous studies have been conducted in this field to compare various ML-based techniques. However, the existing literature still lacks a comprehensive examination, especially considering different OSN platforms. Additionally, the utilization of bio-inspired algorithms has been largely overlooked.

Our study takes a novel approach by conducting an extensive comparison analysis of various fake profile detection techniques in online social networks. The results of our study indicate that supervised models, along with other machine learning techniques, as well as unsupervised models, are effective for detecting false profiles in social media. To achieve optimal results, we have incorporated six bio-inspired algorithms to enhance the performance of fake profile identification.résultats .

Mots clé : Online Social Network , fake profile ,detection,machine learning.

Contents

List of Figures

List of Tables

List of acronyms

Abbreviation	Acronyme
BI	Bio-Inspired
ML	Machine Learning
DT	Decision Trees
RF	Random Forest
LR	Linear Regression
SVM	Support vector machine
KNN	k-nearest neighbor
GOW	Grey Wolf Optimizer
MFO	Moth-flame Optimization
GOA	Grasshopper Optimization Algorithm
WOA	Whale optimization algorithm

Chapter 1

Introduction

One of the most popular applications in the mobile device is the Online Social Network (OSN),is an essential element to connect people around the earth for sharing various data items includes videos, photos, and messages .Online social networks OSN have been growing sharply over the last century due to technology's growth. Nowadays, about 4.66 billion people worldwide use the Internet, and 4.14 billion are active users on social media [9].

1.1 Problem statement

The number of fake accounts (or profiles) is growing dramatically, causing massive problems and leading to misbehavior, including political, fake-news-spreading, blackmailing, misleading ads, terrorist propaganda, spam, and hate speech. There are various categories of fake profiles, including compromised profiles, cloned profiles, and online bots (spam bots). fake accounts are also known as "fake profiles," and they are a major security and privacy concern for OSNs protection. There has been an unexpected increase in the number of registered users on popular websites like Facebook, Twitter, LinkedIn, Instagram, and others, many of whom are fake and were made for particular purposes.[18].

As a result, these social networking platforms' credibility and reputation took a significant hit.

1.2 Current literature and motivation

One of the primary issues with OSNs is ensuring user security and privacy against fake profiles. Recently, researchers have used Machine Learning (ML) techniques to automate and improve the detection of bogus accounts, like K-Nearest Neighbour, Support Vector Model, Decision tree, Random forest.. etc

Because information is publicly available by default on Twitter, it has been the most popular target among OSNs in recent studies, surveys, and literature reviews to investigate and compare various ML-based techniques. Furthermore, few new works have addressed the application of bio-inspired algorithms to improve the presented methodologies.

1.3 Contribution and results

In order to contribute to the current research, we conducted a new comparative study of various fake profile detection algorithms in online social networks (OSNs) in this paper. Our research considers several OSN sites, including Facebook, Twitter, and Instagram. Additionally, we explore bio-inspired algorithms. Our research's goal is not to choose the best fake profile detection strategy out of all the existing ones, but rather to select the best ones in each category of app.

The results of our work suggest that fake profile identification in social media is suitable for supervised models and may be improved using various machine learning approaches like k-cross validation and parameter tweaking, as well as for unsupervised models like k- means. Add to

that, the good results provided by many bio-inspired algorithms, which have proven its works compared to the old algorithms.

1.4 Dissertation structure

The dissertation is divided into four chapters. Chapter 2 offers the necessary back-ground for the dissertation and places it within the third chapter of state-of-the-art. The methods we used to perform our comparative analysis is presented in Chapter 4. Finally, Chapter 5 provides the acquired results and discusses them.

Chapter 2

Basic concept

2.1 Introduction

We all know that social media has become a huge part of our lives and is our main need. They are great platforms to share information and interact with people But now we know there are two sides to the coin, let's worry about social media Abuse stems from behaviors like social media manipulation, one of the best examples on social media Manipulation refers to the creation of fake profiles used for spam, phishing, dissemination of false information and possible identity theft, encouraging social networks to improve their cybersecurity.

Keywords Online social network,Fake profile,Machine learning,metaheuristics .

2.2 Online Social Networks

2.2.1 Definition:

A social network is a social structure made up of individuals or organizations related to one or more people,More types of interdependence (friendship, common interests, work, knowledge, prestige, etc.) 'These are the 'nodes' of the network. Social media are web-based communication tools that facilitate people with creating customized user profiles to interact and share

information with each other. [18]

Web 2.0 : is the network as platform, spanning all connected devices; Web 2.0 applications delivering software as a continually–updated service that gets better the more people use it, remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects and deliver user experiences. [19]

Online Social Networks (OSNs) Online Social Networks are popular applications for sharing various data, including text, photos, and videos.In another term it is meeting people and discovering new opportunities by making a digital arena. All users overall the world have the possibility to express their opinion about several subjects related with politics, education, travel,culture, commercial product or general interests . Beside knowing their feeling as they are expressed by their messages, posts, comments in various platforms. So social network is an important element for the estimation of people's opinion about a particular subject. Examples of OSN include Facebook, Twitter, linkedIn, and Flickr. [59]

2.2.2 Fake profile problem

Fake profile : Fake profiles can be classified into malicious and non-malicious accounts. Non-malicious accounts create a fake profile simply with the purpose of having multiple accounts[46]

Fake profiles are created by stealing the data such as profile name, profile photo, age, sex, and others. This results in exposing incorrect information to their friends and contacts, which are connected to them through social media. This situation can result in huge damage in the real world [55]

The main characteristics of fake profile are :

- It has less account age.
- Small number of followers.
- Not often active.
- Location IP is not provided.
- Location not specified.

2.3 Machine Learning

Machine learning is a subfield of computer science concerned with building algorithms that, to be useful, rely on a set of examples of certain phenomena. Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.[4]

¨A branch of artificial intelligence known as" machine literacy"(ML) is a general term for when a computer learns from data without being expressly tutored to acclimatize to a changing terrain.

2.3.1 Machine Learning history

Machine learning belongs to the crossroad of cybernetics (control science) and computer science. It is attracting recently an overwhelming interest, both of professionals and of the general public. In the talk a brief overview of the historical development of the machine learning field with a focus on the development of mathematical apparatus in its first decades is provided.[17]

In 10 years, machine learning has conquered the industry: today it is at the heart of the magic of high-tech products, ranking web search results, powering smartphone voice recognition and recommending videos, beating the world champion at the game , The ultimate goal of machine

learning is to design algorithms that automatically help a system to learn , grow, change,and improve by themselves, by being specifically programmed.[17]

How machine learning different from AI

Artificial intelligence : Is concerned with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction.[42]

AI is the grand vision of intelligent machines while ML consists of the models ,processes and supporting technology that we've using to get there .

ML is generally considered to be a subset of AI , however these two terms are used interchangeably .

2.3.2 Taxonomy of Machine learning models :

This subsection explains the detailed taxonomy of machine learning-based anomaly intrusion detection Techniques illustrating supervised and unsupervised ML methods

- **Supervised learning** : When an algorithm generates a function that establishes a mapping between inputs and desired outputs, it can be classified into classification and regression methods. Decision tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN), Naive Bayes (NB), and others are well-known algorithms used in supervised learning. One standard formulation of the supervised learning task is the classification problem

- **Unsupervised learning**: A set of inputs is being modeled, where labeled examples are unavailable. Unsupervised machine learning focuses on acquiring a function that explains an unknown condition based on unlabeled data. Common unsupervised algorithms comprise

K-means, Self-Organizing Maps (SOMs), hidden Markov model (HMM), and Autoencoders (AEs).

- **Semi-Supervised Machine Learning**: which combines both labeled and unlabeled examples to generate an appropriate function or classifier

Different between Supervised learning and Unsupervised learning: There is no universally accepted method to assess the precision of unsupervised learning algorithms, primarily because of the absence of labeled data. This fundamental distinction sets them apart from supervised learning algorithms

Supervised models

Decision tree

Decision tree belongs to supervised learning algorithms. A decision tree is a popular classification method that generates tree structure where each node denotes a test on an attribute value and each branch represents an outcome of the test 'see figure' 2.1 [1])

The concept involves dividing the data space into regions of high density and low density. This algorithm is based on statistical principles, where attributes are chosen in a tree-like structure, starting from the root and progressing towards the leaves.

[1]https://www.datacamp.com/

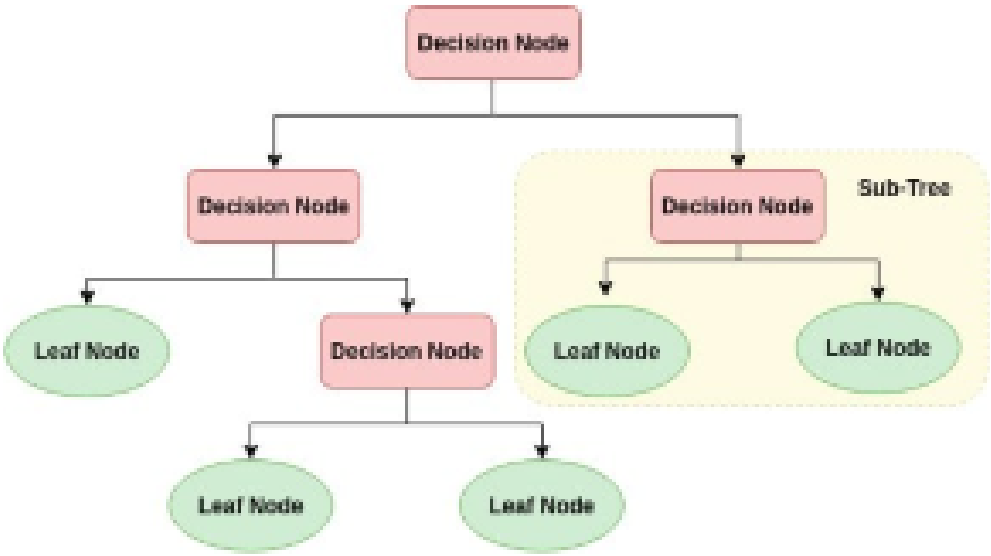


Figure 2.1: *Simple representation of decision tree*

K-nearest neighbors (KNN) The k-nearest neighbors (KNN) algorithm is a supervised machine learning algorithm used to solve both classification and regression problems,but especially in the classification.

The definition of nearest neighbors is based on the computation of the Euclidean distance from the new data point to each of the existing data points. The Euclidean distance is the most common distance measure [31].

The letter k is used to indicate the number of neighbors to use. To compute the k nearest neighbors, you simply compute the distance between your new data point and each of the data points in the training data. Depending on which number you have for k, you take the k data points that have the lowest distance.[31]

One of the drawbacks of K-Nearest Neighbors is that it is sensitive to inconsistent data (noisy) and missing value data. The figure 2.2 below shows how the knn classifier works.

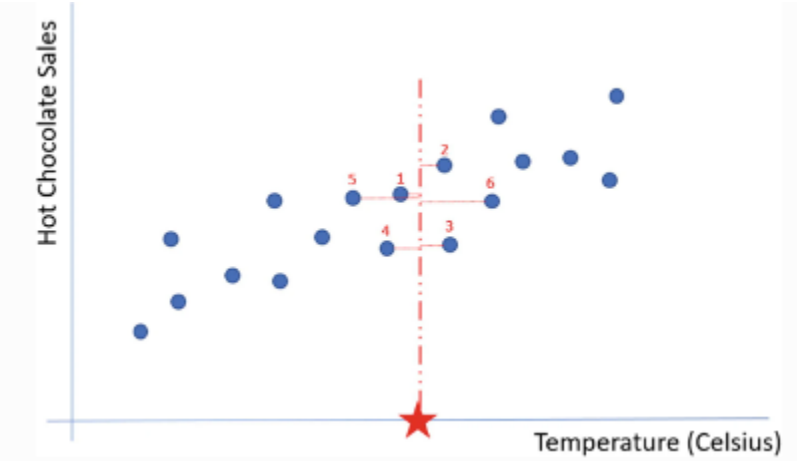


Figure 2.2: *KNN Classifier [31]*

Random Forest Machine Random Forest is a machine learning algorithm that considered as a supervised learning technique. It creates several Decision Trees on the subset of data.

Moreover, Random Forest is used in Regression and Classification of ML. It is proved the effectiveness of this algorithm on large datasets compared to other classifiers like: Neural Networks, Discriminant Analysis and Support Vector Machines (SVM)[39]

One of the most important benefits of Random Forest is that it can work with missing data, which is the relief of missing values by the variable that's common in a particular knot. The Random Forest can also handle big data snappily, give a advanced delicacy and help over-fitting problems. One the other hand, Random Forest requires numerous computational ressources and large memory for storehouse, due to the fact that it creates a lot of trees to save information piped generated from hundreds of individual trees.[45]

The figure 2.3 shows how Random Forest works.

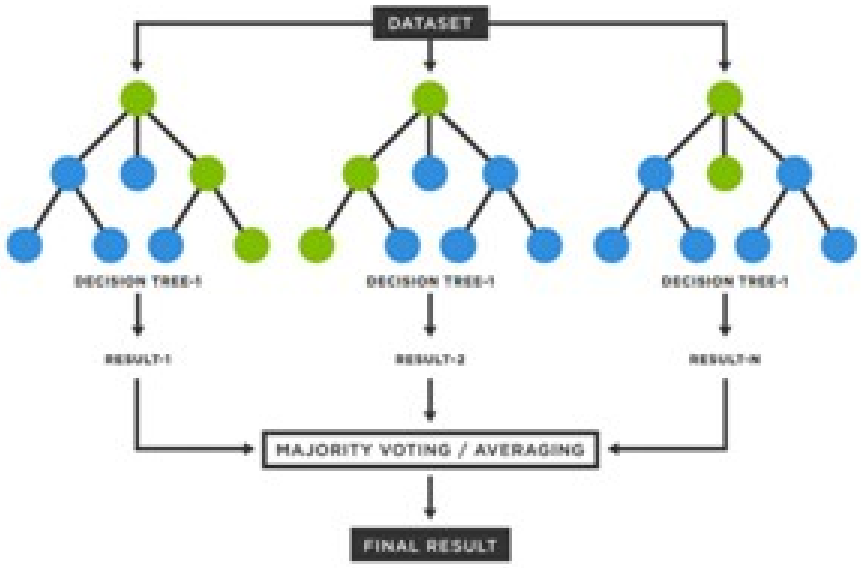


Figure 2.3: *Random Forest Classifier [45]*

Support Vectors Machine

Support Vector Machines (SVMs) are supervised learning models utilized for both classification and regression tasks. They are capable of addressing linear and nonlinear problems by leveraging the concept of Margin to distinguish between different classes.

SVMs give better accuracy than KNNs, Decision Trees, Naive Bayes Classifiers in most cases and have been known to outperform neural networks in a few instances.

The support vector machine algorithm's objective is to find a hyperplane in an N-dimensional space that distinctly classifies the data points and to find the optimal separating hyperplane or maximum-margin hyperplane, which separates the N different data points clusters [32].

- The support vectors represent the data points that reside on or are in proximity to the hyperplane. These points play a significant role in determining the position and orientation of the hyperplane.

- Hyperplanes are boundaries utilized to make decisions and classify the data points

The figure 2.4 shows the representation pf hyper planes

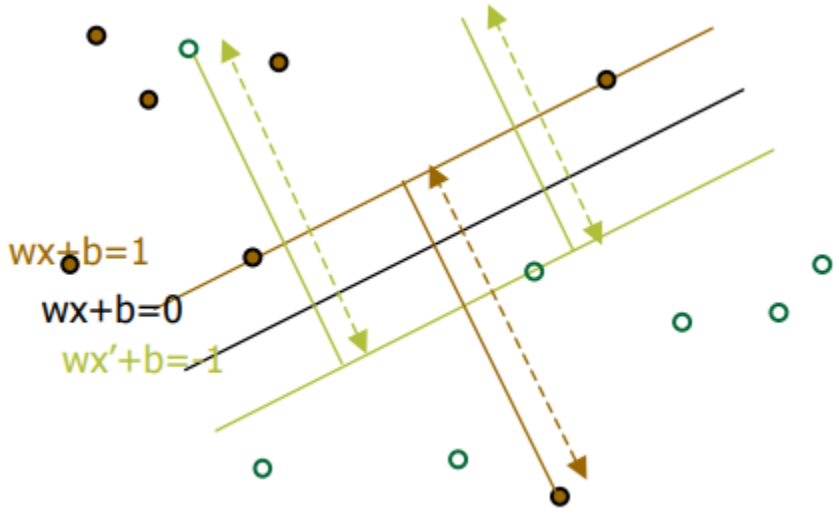


Figure 2.4: *Representation of Hyper planes [25]*

Unsupervised model

Kmeans algorithm

K-means clustering stands out as a straightforward and well-liked unsupervised machine learning algorithm .

In the context of attribute problems, each instance is represented in an m-dimensional space. The cluster centroid, which serves as a representation of the cluster, is a point in the m-dimensional space that encapsulates the instances belonging to the cluster.

The typical measure of distance between an instance and a cluster center is often used the Euclidean distance though variations such as the Manhattan distance (step-wise distance) are common. As most implementations of K-Means clustering use Euclidean distance [11].

- A cluster denotes a group of data points that are gathered or consolidated together .

- K represents the desired number of centroids that are required in the dataset.
- A centroid is the theoretical or actual position that represents the central point of a cluster.

The k-means clustering algorithm operates through three primary steps to determine the similarity between items and group them into clusters.

- Select the k values.
- Initialize the centroids.
- Select the group and find the average.

The figure 2.5 shows the flowchart of kmeans clustering

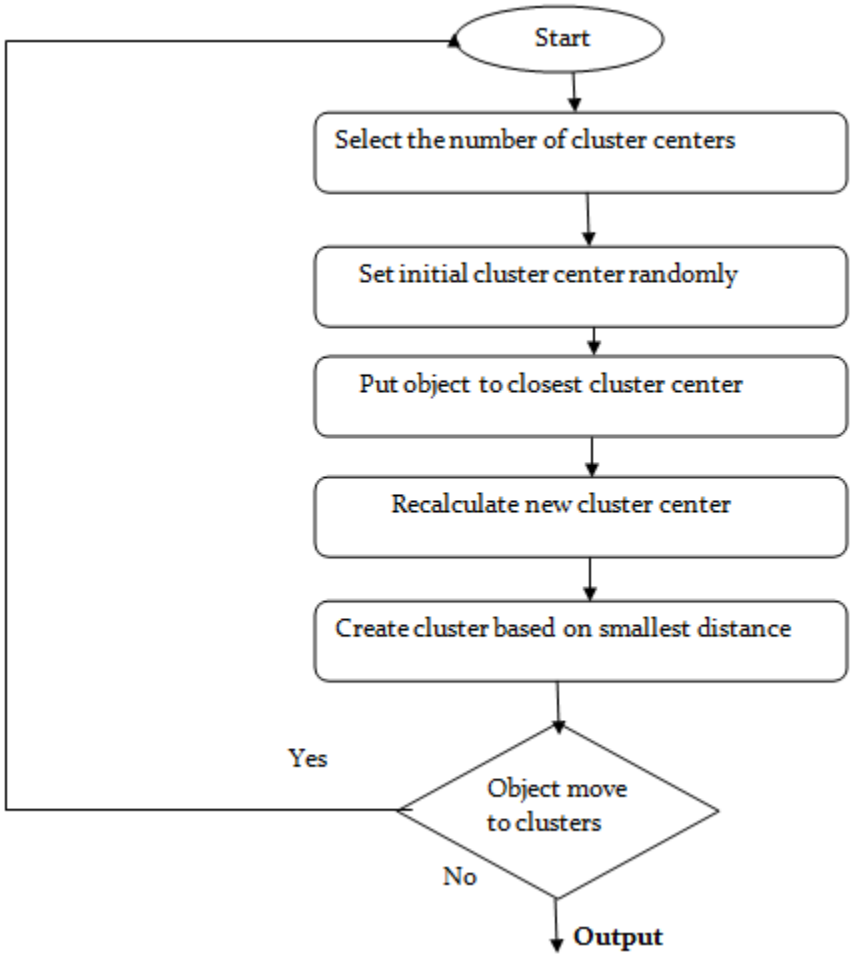


Figure 2.5: *Flowchart of k-means clustering algorithm [56]*

2.4 metaheuristics

2.4.1 definition

Metaheuristics, which are frequently inspired by nature, are versatile algorithmic frameworks intended for tackling intricate optimization problems.

In contrast to precise methods, metaheuristics offer effective solutions in a timely manner, making them suitable for managing large-scale problem scenarios. However, they do not guarantee the discovery of global optimal solutions or even constrained solutions.

Metaheuristics have received more and more popularity in the past 20 years. Their use in many applications shows their efficiency and effectiveness to solve large and complex problems.[13]

2.4.2 Some areas of metaheuristics

- Metaheuristics are employed in various fields, including but not limited to engineering design, topology optimization, structural optimization in electronics and VLSI, aerodynamics, fluid dynamics, telecommunications, automotive, and robotics

- Metaheuristics find applications in system modeling, simulation, and identification across disciplines such as chemistry, physics, and biology. Additionally, they are utilized in control systems, signal processing, and image processing.

- Planning in routing problems , robot planning , scheduling and production problems , logistics and transportation , supply chain management , environment and so on.[13]

2.4.3 Classification of meta-heuristic algorithms

Several approaches have been suggested for categorizing meta-heuristics, depending on the chosen characteristics.

Nature-inspired against non-nature inspired: Meta-heuristics can be classified into different categories based on the algorithm's origin. The majority of these methods are nature-inspired algorithms, including Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Genetic Algorithms (GA). Additionally, there are non-nature-inspired algorithms such as Iterated Local Search (ILS).[10]

Population-based against single point search:

Meta-heuristics can also be classified based on the number of solutions employed simultaneously. Trajectory methods, which consist of local search-based meta-heuristics such as TS, ILS, and Variable Neighborhood Search (VNS), operate on a single solution at a time. In contrast, population-based algorithms, similar to swarm-based meta-heuristics, conduct parallel searches with multiple initial points.[60]

Dynamic against static objective function: Classification can also be based on how the objective function is utilized in meta-heuristics. Some algorithms maintain the objective function in its original form within the problem representation, while others, such as Guided Local Search (GLS), modify it dynamically during the search. The underlying concept behind this approach is to avoid getting trapped in local optima by altering the search landscape. Consequently, the objective function is adjusted by integrating the gathered information throughout the search process.[60]

Various against single neighborhood structure

Most meta-heuristic algorithms utilize a single neighborhood structure, meaning that the topology of the fitness landscape remains unchanged throughout the algorithm. However, there are algorithms such as Variable Neighborhood Search (VNS) that employ multiple neighborhood structures. This flexible structure allows for diversifying the search by switching between different fitness landscapes, thereby enhancing exploration capabilities.[60]

Memory usage against memory-less methods The utilization of memory is a crucial factor in classifying meta-heuristics. In other words, the ability to effectively use memory is considered a fundamental aspect of a powerful meta-heuristic. Memory-less algorithms operate as Markov processes, where the next action is determined solely by the current state of the search process.

There are various approaches to employing memory, and the use of short-term memory often differs from long-term memory. Short-term memory typically keeps track of recent moves, visited solutions, or decisions made. On the other hand, long-term memory usually involves the accumulation of synthesized parameters related to the search.[60]

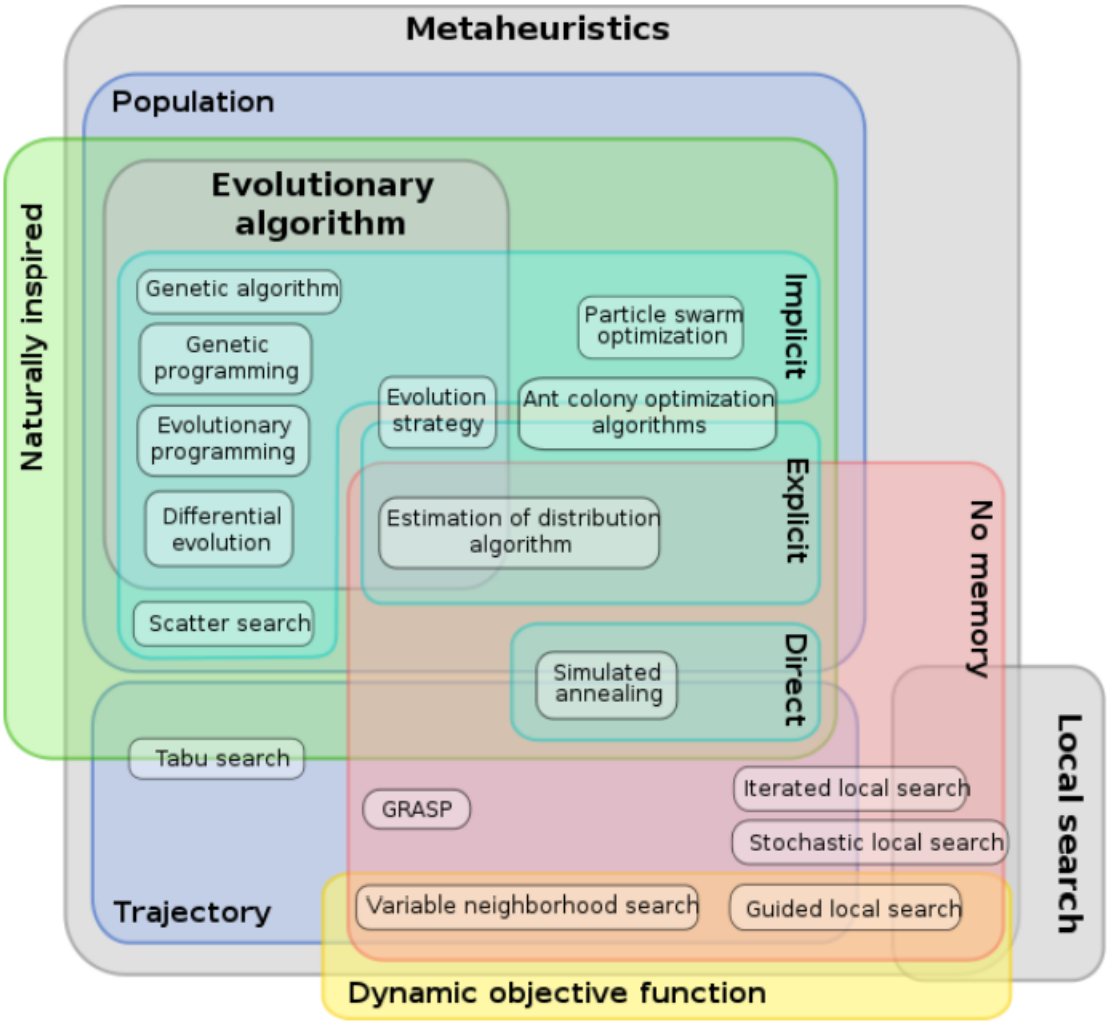


Figure 2.6: *Classification of meta-heuristic algorithms [60]*

2.5 Bio-inspired algorithms

Bio-Inspired Computing lies within the realm of Natural Computing, is a field to perform research that is concentrated on both biology as an inspiration for solving complex computational problems and the use of the natural world experiences to solve real world problems. It is a field which is related Biology, Computer Science, Informatics, Cognitive Science, and robotics.

The main advantages of bio inspired computing are robust, scalable, adaptable, flexible, and also decentralized, with many relatively simple individual units that act and interact locally, as a global information processing and coordination.[34]

2.5.1 Bio-inspired algorithms (BIAs) taxonomy

Bio-inspired algorithms have proven to be highly effective approaches for tackling complex optimization problems and have been successfully employed to solve a wide range of problems across different domains. Throughout the last few decades, numerous Bio-Inspired Algorithms have been developed, drawing inspiration from diverse biological swarms found in nature.[49]

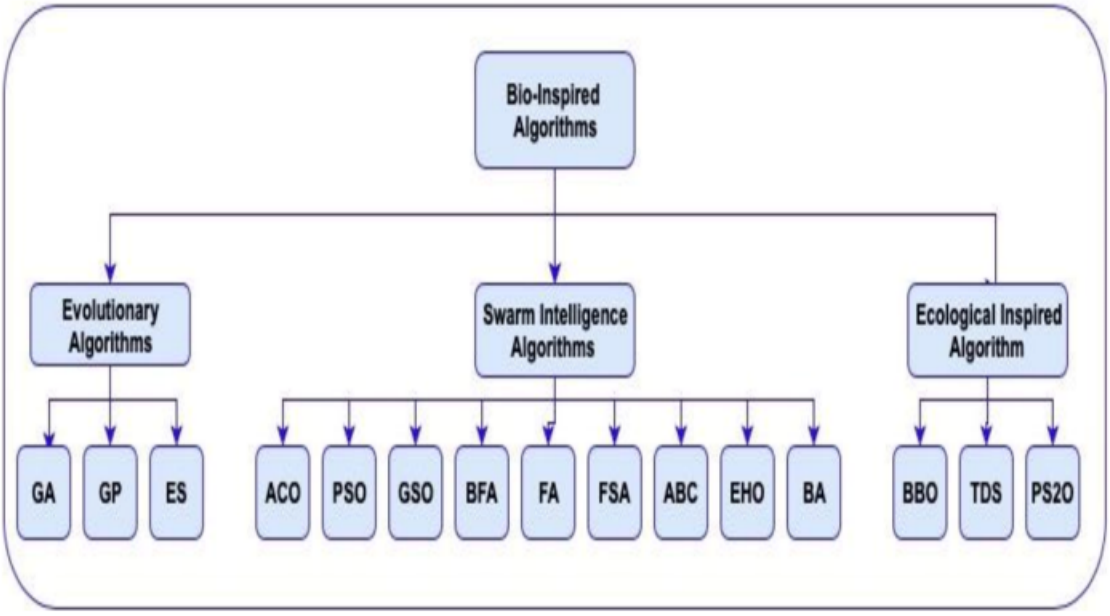


Figure 2.7: *Bio-Inspired Algorithm Taxonomy. [49]*

Meaning of the Abbreviations :

Abbreviation	Meaning
GA	Genetic Algorithm
GP	Genetic Programming
ES	Evolutionary Strategy
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
GSO	Glowworm Swarm Optimization
BFA	Bacterial Foraging Algorithm
FA	Firefly Algorithm
FSA	Fish School Algorithm
ABC	Artificial Bee Colony
EHO	Elephant Herding Optimization
BBO	Biogeography Based Optimization
TDS	Temperature Dependent Sex
PS2O	Symbiosis

2.6 Some Bio-inspired Algorithm utilizing in our work

2.6.1 Grey wolf optimizer algorithm

History: In 2014, Seyedali Mirjalili introduced the Grey Wolf Optimizer (GWO) algorithm, which takes inspiration from the social behavior and hunting strategies of grey wolves. Mirjalili's objective in developing the GWO algorithm was to design a nature-inspired optimization technique capable of effectively solving complex optimization problems. By observing the hierarchical structure and cooperative behavior displayed by grey wolves in their packs, Mirjalili translated these principles into an optimization algorithm.[50]

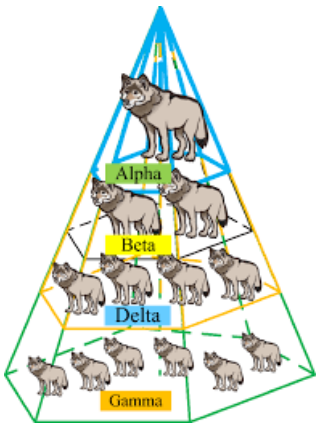


Figure 2.8: *The Grey Wolf Optimize*

Principal: The Grey Wolf Optimizer (GWO) algorithm operates by emulating the behavior of the alpha, beta, and delta wolves within a pack, which represent the top individuals. These leading wolves guide the rest of the pack (referred to as omega wolves) towards improved solutions. In essence, the GWO algorithm leverages the hierarchical dynamics observed in wolf packs to guide the search process and converge towards optimal solutions.[23] Here are the main steps of the GWO algorithm:

- **Initialization:** The pack is represented by a set of wolves (potential solutions). Each wolf is associated with a position in the search space.
- **Evaluation:** Each wolf is evaluated using an objective function that measures its quality with respect to the given optimization problem
- **Position update:**The alpha, beta, and delta wolves are identified among the pack based on their evaluations. Then, the positions of the wolves are updated using formulas that simulate the hunting and movement behavior of wolves in the search space.
- **Exploration and exploitation:** The alpha, beta, and delta wolves guide the other omega

wolves to explore the search space. The new positions of the omega wolves are updated by combining the movements of the alpha, beta, and delta wolves.

- **Termination criterion:** The algorithm continues to iterate until a predefined termination criterion is met, such as reaching the maximum number of iterations or achieving a sufficiently small improvement in the solution.[23]

The core concept behind the Grey Wolf Optimizer (GWO) is that the alpha, beta, and delta wolves embody the current best solutions found so far. The remaining wolves in the pack then align their positions and behaviors towards these superior solutions using specific mathematical equations. This alignment mechanism allows the wolves to converge towards better solutions as the optimization process progresses

The GWO algorithm has been successfully applied to numerous optimization problems in various domains, such as engineering, finance, machine learning, etc. It provides an effective alternative for finding global solutions to complex problems by leveraging the principles of nature[23]

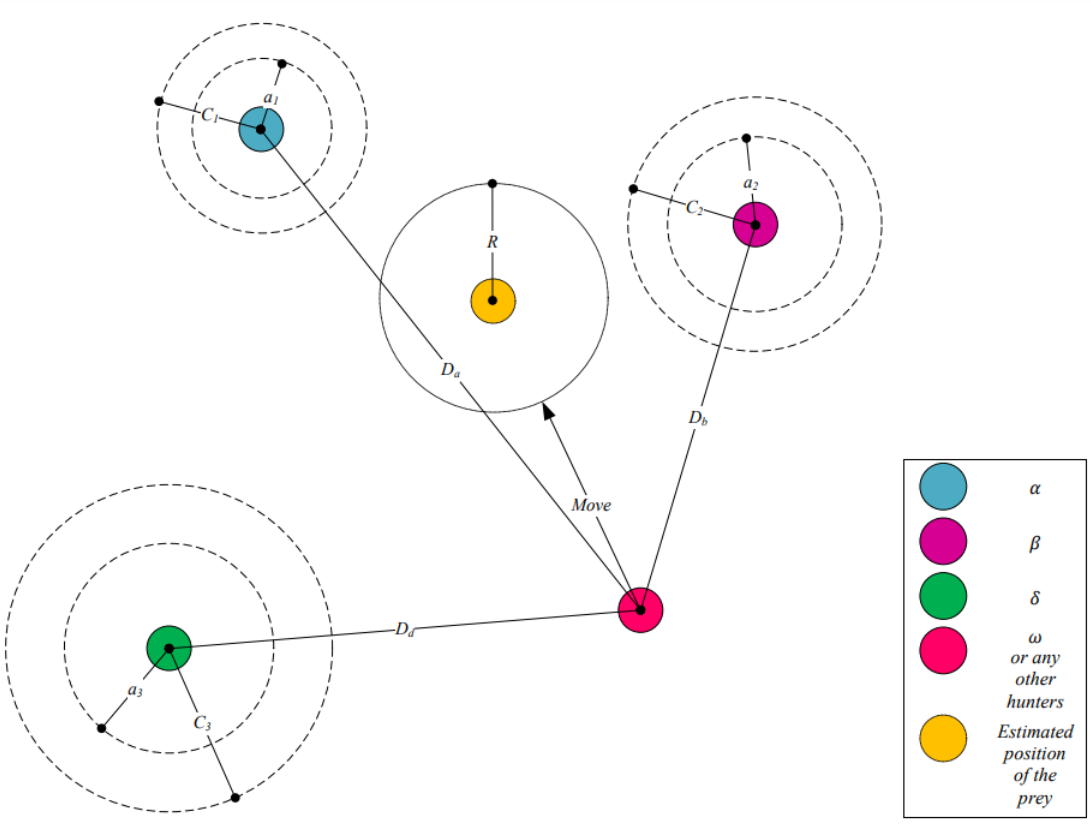


Figure 2.9: *Position updading in GWO*

Pseudo-code:

Algorithm 1 GWO Algorithm

Initialize of grey wolves ω_n where n= 1,2, ...,N according to given upper bound (U B) and lower bound (L B) values
Initialize a , A and C .
Evaluate the fitness of all search agents .
Select alpha , beta and delta as :
ω_α=best search agent
ω_β=second best search agent
ω_δ=third best search agent
Initialize i =0 and Max_it = Maximum number of iterations allowed .
while i<Max_it **do**
    for each search agent do
        Update the position of current search agent according to (7)
    end for
    Update a , A and C.
    Evaluate the fitness of all search agents.
    Update ω_α ,ω_β and ω_δ.
    i=i+1.
end while
Return ω_α

2.6.2 Whale optimization algorithm

History: Whale Optimization Algorithm (WOA) is another SI-based optimization method inspired by the hunting behavior of humpback whales (Mirjalili and Lewis, 2016).[36]

The WOA (Whale Optimization Algorithm) algorithm was developed with the intention of designing a nature-inspired optimization technique capable of effectively solving complex optimization problems. This algorithm draws inspiration from the hunting strategies and cooperative behavior observed in humpback whales. By emulating the characteristics and behaviors of these whales, the WOA algorithm aims to provide a robust and efficient approach to optimization.[33] The WOA algorithm gained attention due to its simplicity, effectiveness, and

ability to handle a wide range of optimization problems. It quickly became a promising alternative to traditional optimization algorithms. The algorithm simulates the hunting behavior of humpback whales, which involves individual and group movements to locate and trap prey. The WOA algorithm employs a set of equations and rules that imitate the hunting process of the whales, guiding the search for optimal solutions.[33] Since its introduction, the WOA algorithm has been applied to various fields and domains. Researchers have utilized it in engineering design problems, image processing, data mining, feature selection, and other optimization tasks. Its ability to find near-optimal solutions and its convergence properties make it a valuable tool for solving complex optimization problems. Over time, researchers have proposed enhancements and modifications to the WOA algorithm to improve its performance and adaptability to specific problem domains. These variations include modified equations, hybridizations with other algorithms, and parameter tuning approaches.[33] The Whale Optimization Algorithm continues to be an active area of research, with ongoing efforts to refine its efficiency, scalability, and applicability to a broader range of optimization problems. Its historical development and subsequent advancements have established it as a notable algorithm in the field of nature-inspired optimization. [33]

Figure 2.10: *The Whale Optimization Algorithm*

Principal:

The Whale Optimization Algorithm (WOA) is a nature-inspired optimization algorithm that emulates the hunting behavior of humpback whales. By imitating the movement patterns and cooperative strategies employed by whales during their hunting process, WOA aims to effectively solve optimization problems.[37]

Here are the main steps of the WOA algorithm:

- **Initialization:** The algorithm starts by initializing a population of candidate solutions, often referred to as "whales." These solutions represent potential solutions to the optimization problem.

- **Exploration and Exploitation:** The WOA algorithm employs two main phases to balance exploration and exploitation. In the exploration phase, whales move randomly in search of potential prey (optimal solutions) within the search space. In the exploitation phase,whales tend to converge toward the most promising solutions they have discovered

so far.

- **Updating Positions:** The positions of the whales are updated iteratively based on specific equations that mimic the movement patterns of whales. The equations govern the movement towards better solutions while maintaining a balance between exploration and exploitation.

- **Fitness Evaluation:**After updating the positions, the fitness values of the whales are evaluated using an objective function. The objective function measures the quality or performance of each solution based on the problem being optimized.

- **Dominance and Selection:** The WOA algorithm utilizes dominance and selection techniques to determine the most promising solutions. Dominance is used to compare and rank the solutions based on their fitness values, while selection ensures that the better solutions are retained for further exploration and exploitation.

- **Termination Criterion:**The algorithm continues iterating through the exploration, exploitation, position updating, fitness evaluation, and selection phases until a termination criterion is met. This criterion can be a maximum number of iterations, a desired level of solution quality, or a convergence measure indicating that further iterations are unlikely to significantly improve the solutions[37]

The WOA algorithm utilizes the cooperative behavior and exploration-exploitation balance observed in humpback whales to efficiently search for optimal solutions. Through iterative updates of whale positions and fitness evaluations, the algorithm aims to converge towards near-optimal solutions within the search space.[37]

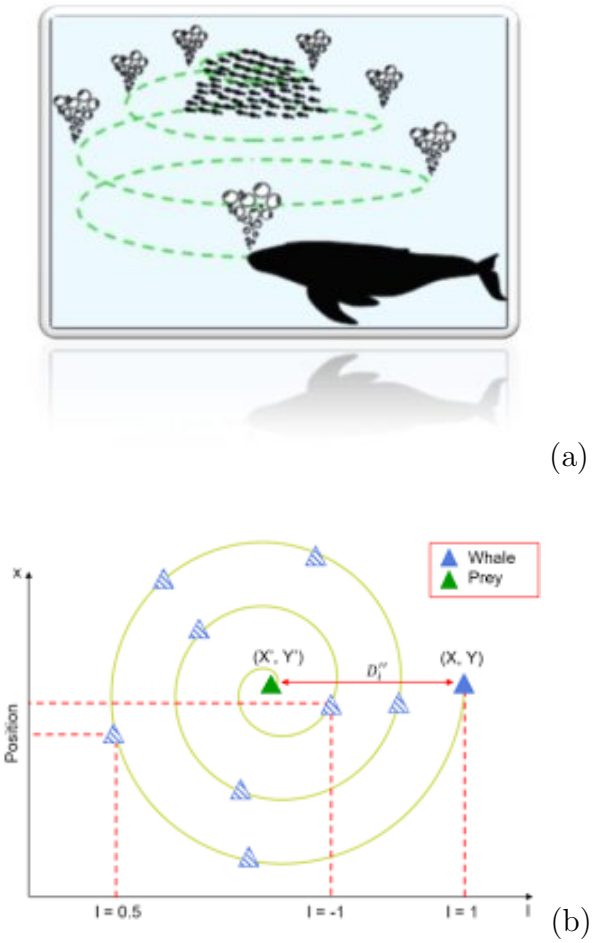


Figure 2.11: *Graphical abstract of Whale Optimizer*

Pseudo-code:

Algorithm 3 WOA Algorithm

Initialize the whales population X_i(i=1,2,3,.....,n)
Initialize a ,A and C.
Calculate the fitness of each search agent
X^*= the best search agent
procedure WOA(Population, a, A,C,MaxIter,..)
    i=1
    while i<MaxIter **do**
        for each search agent do
            if |A| <=1 **then**
                Update the position of the current search agent by the equation 2.6
            else
                if |A|>=1 **then**
                    Select a random search agent X_r and Update the position of the current agent by the equation 2.8
                end if

                Update a,A,and C
                Update X^* if there is a better solution .
                i=i+1.

                Return X^*

2.6.3 Grasshopper Optimization Algorithm

History: The grasshopper optimization algorithm (GOA) is a meta-heuristic algorithm proposed in 2017 mimics the biological behavior of grasshopper swarms seeking food sources in nature for solving optimization problems. Algorithms Grasshopper Optimization Algorithm (GOA) is a recent algorithm proposed by Saremi et al. that belongs to the family of Swarm Intelligence techniques. This algorithm mimics the navigation of adult grasshopper in nature when forming one of the largest swarms on the planet[53]

Some of the main advantages of this algorithm are: a small number of controlling parameters, adaptive exploratory and exploitative search pattern, and gradient-free mechanism.

The algorithm gained attention for its simplicity and effectiveness in handling a wide range of optimization problems. It offers an alternative approach to traditional optimization algorithms and has shown promising results in terms of convergence and solution quality.[52] Since its introduction, the GOA algorithm has been applied to various fields and domains. Researchers have utilized it in engineering design problems, data mining, feature selection, image processing, and other optimization tasks.

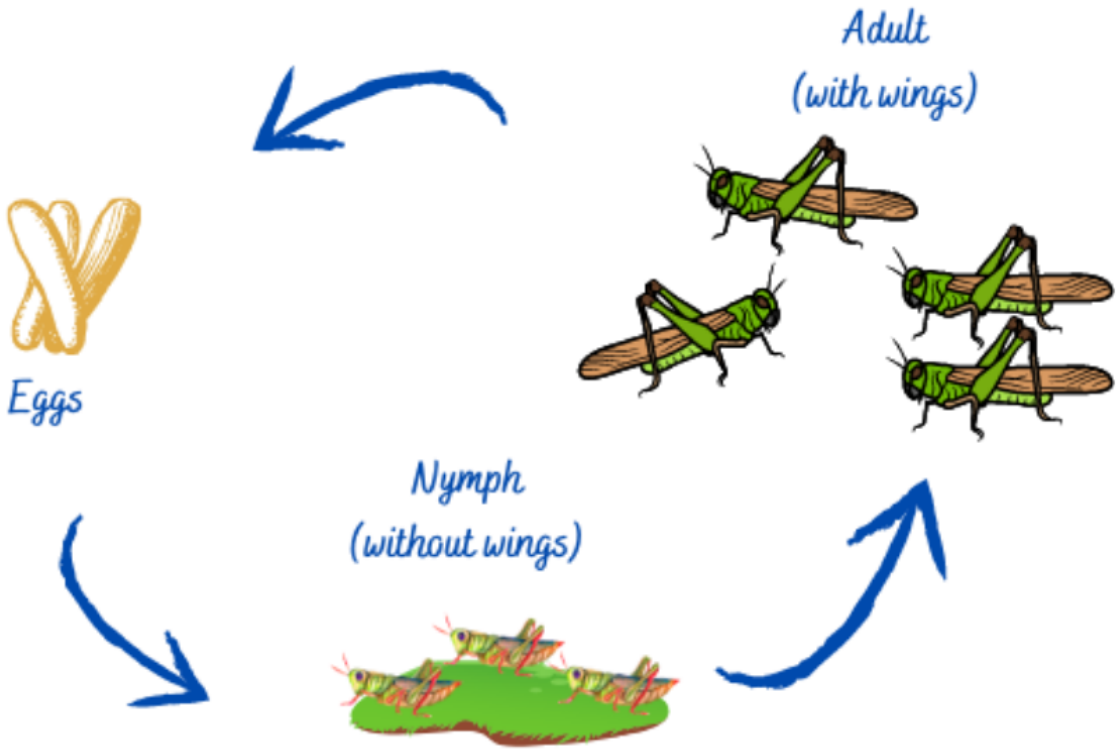


Figure 2.12: *The Grasshopper Optimization Algorithm*

[**?**]

Principal: The Grasshopper Optimization Algorithm (GOA) is an optimization algorithm that draws inspiration from the swarming and jumping behavior of grasshoppers. It seeks to solve optimization problems efficiently by simulating the movement and interaction of these insects. The algorithm operates based on the following core principles:[52]

- **Initialization:** Algorithm (GOA) begins with the initialization of a population of grasshoppers, which serve as potential solutions to the optimization problem. The grasshoppers' initial positions are randomly distributed within the search space

- **Fitness Evaluation:** The quality or performance of each grasshopper solution in the Grasshopper Optimization Algorithm (GOA) is evaluated through the use of an objective function. This objective function quantifies the fitness value of each grasshopper, indicating how well it performs in solving the given optimization problem.

- **Movement and Interaction:** The grasshoppers' movement is guided by their position and the influence of other grasshoppers. Each grasshopper adjusts its position based on its own experience and the best solution found so far. The movement is determined by a combination of local and global search strategies.

- **Local Search:** During the local search phase, each grasshopper in the Grasshopper Optimization Algorithm (GOA) explores its immediate neighborhood to seek out improved positions. This process enables the algorithm to exploit and focus on promising regions within the search space.

- **Global Search:** In the global search phase of the Grasshopper Optimization Algorithm (GOA), grasshoppers interact with each other by sharing information about their best

positions. They adapt their movements to converge towards better solutions discovered by other grasshoppers. This interaction promotes exploration of diverse regions within the search space, facilitating the convergence towards optimal or near-optimal solutions.

- **Update:** The positions of the grasshoppers are updated iteratively based on their movement and interaction. The algorithm aims to improve the quality of solutions over time by guiding the grasshoppers towards better regions of the search space.

- **Termination Criterion:** The algorithm continues iterating through the movement, interaction, local search, and global search phases until a termination criterion is met. This criterion can be a maximum number of iterations, a desired level of solution quality, or a convergence measure indicating that further iterations are unlikely to significantly improve the solutions. By combining local and global search strategies and facilitating the interaction among grasshoppers, the Grasshopper Optimization Algorithm aims to efficiently explore the search space and converge towards optimal or near-optimal solutions.[53]

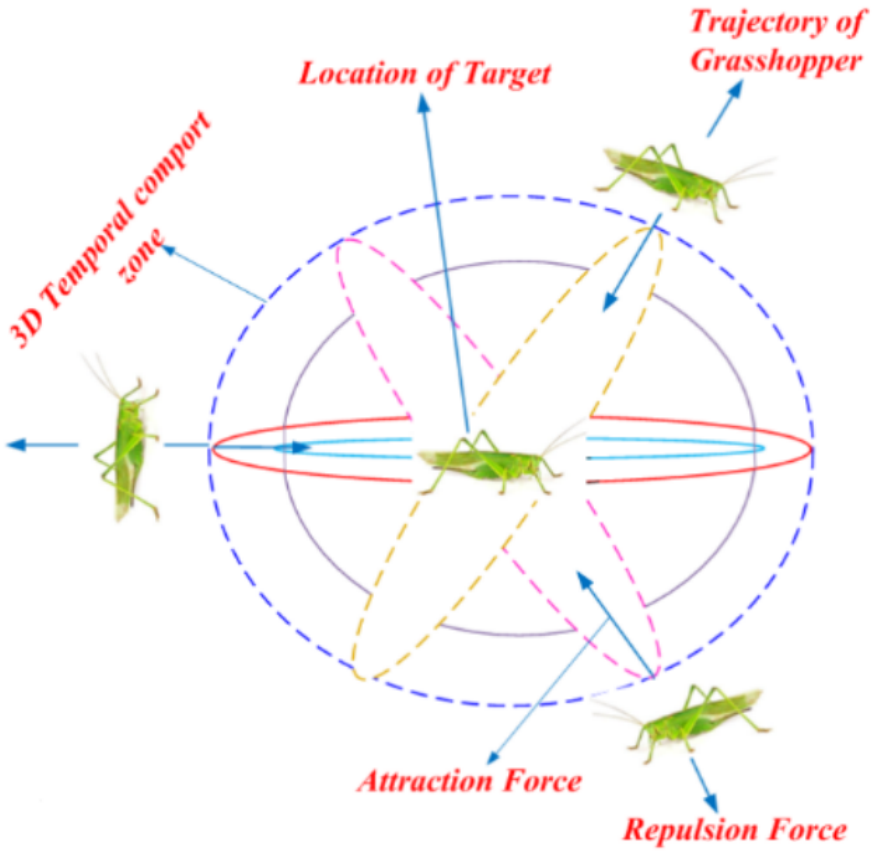


Figure 2.13: *Graphical abstract of Grasshopper Optimization Algorithm*

Pseudo-code:

Algorithm 1: Grasshopper Optimizer

Parameter Initialization: iter, $c_{max}, c_{min}, l,$ and f
Swarm Initialization $X_i (i = 1, 2, 3, ..., n)$
Compute Fitness value for each Grasshopper (search agent)
Select the best solution among all (best search agent)
while *(current iteration (iter) ¡ maximum iteration (Max_{iter}))* **do**
    Update c using Equation 9
    for *each grasshopper* **do**
        Normalize distance between grasshopper in the range [1, 4]
        Update the position of grasshopper using Equation 8
        Bring current grasshopper back if it goes outside the boundaries
    end
    Update the best solution if there's a better one
    iter = iter + 1
end
Return the best solution

2.6.4 Moth flame optimizer

History: The Moth Flame Optimizer (MFO) algorithm was introduced by Seyedali Mirjalili, Mohammad Salimi, and Andrew Lewis in 2015. Inspired by the natural behavior of moths being attracted to light sources at night, the algorithm aims to solve complex optimization problems. By imitating the navigation and attraction behavior of moths towards a light source, the MFO algorithm offers a nature-inspired optimization technique for effectively optimizing given problems.

The Moth Flame Optimizer (MFO) algorithm has garnered attention for its simplicity and versatility in tackling diverse optimization problems. It provides an alternative approach to conventional optimization algorithms and has demonstrated promising outcomes in terms of convergence and solution quality. The algorithm's historical development and ongoing re-

search endeavors have contributed to its increasing popularity as a nature-inspired optimization technique[29]

Figure 2.14: *The Moth Flame Optimizer*

Principal: The MFO algorithm has gained recognition for its simplicity and versatility in addressing a wide range of optimization problems. It presents an alternative methodology to conventional optimization algorithms and has demonstrated promising outcomes in terms of convergence and solution quality. [41] The algorithm involves the following main steps:

- **Initialization:** The Moth Flame Optimizer (MFO) algorithm starts by randomly initializing a population of moths, which represent potential solutions to the optimization problem. The initial positions and brightness values of the moths are set randomly to kickstart the optimization process.

- **Movement towards Light:** The movement of moths in the Moth Flame Optimizer (MFO) algorithm is influenced by their attraction towards a light source, symbolizing an optimal solution. Through iterative updates, the positions of the moths are adjusted based on their

brightness values and distance from the light source. Moths with higher brightness values tend to move closer to the light source, indicating a stronger attraction, while moths with lower brightness values explore different areas of the search space.

- **Neighborhood Attraction:** The Moth Flame Optimizer (MFO) algorithm incorporates a neighborhood attraction mechanism, enabling moths to interact and communicate with one another. Moths have the ability to share information about their positions and brightness values, influencing the movement of their neighboring moths. This interaction promotes cooperation and information exchange, leading to effective exploration and exploitation of the search space.

- **Light Intensity Adjustment:** The light intensity, representing the objective function value, is updated during the optimization process. The light intensity attracts moths and affects their movement towards better solutions

- **Termination Criterion:** The Moth Flame Optimizer (MFO) algorithm iteratively progresses through the movement, neighborhood attraction, and light intensity adjustment phases until a termination criterion is satisfied. This criterion can be defined as a maximum number of iterations, a desired level of solution quality, or a convergence measure indicating that further iterations are unlikely to significantly enhance the solutions.[41]

Since its inception, the MFO algorithm has been widely utilized in diverse domains for solving various optimization problems. Researchers have actively contributed to the algorithm's advancement by proposing modifications and variations to improve its performance and adaptability to specific problem types.[41]

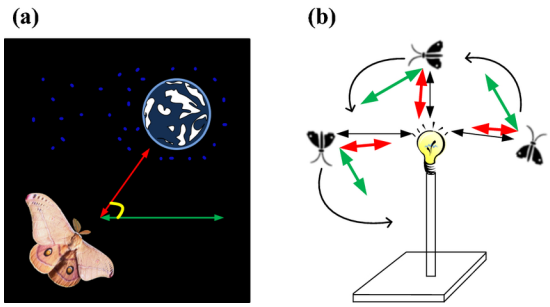


Figure 2.15: *Spiral flying path for moths around close light source*

Pseudo-code:

Algorithm 4 MFO Algorithm

```
Initialize the parameters for Moth-flame
Initialize Moth position M_i randomly
for i=1 to n do
    Calculate the fitness function f_i
end for
while iteration<= Max_iterations do
    Update the position of M_i
    Calculate the number of flames using Eq.()
    Evaluate the fitness function f_i
    if iteration==1 then F=sort(M) and OF=sort(OM)
    elseF=sort(M_{t-1}, M_t) and OF=sort(M_{t-1}, M_t)
    end if
    for i=1 to n do
        for j=1 to d do
            Update the values of r and t
            Calculate the value of D respect to its corresponding moth using Eq. ()
            Update M(i,j)respect ti its corresponding moth using Eq.()
        end for
    end for
end while
Print the best solution =0
```

2.7 Bio inspired computing related to Artificial Intelligence

- Bio-inspired computing is a field of study that is related to the topics of connectionism, social behavior and emergence.

- Biologically-inspired computing is a major group of natural computation.

- The way how this bio-inspired computing differs from artificial intelligence is in how it

implements evolutionary approach to learning, which is opposite to creationist methods used in artificial Intelligence.

- In AI, intelligence is programmed from where the programmer is the creator, creates something and implements it with intelligence.

2.8 Conclusion :

In this chaper , we aim to provide an overview of the fundamental concepts used in our work, including definitions of Online Social Networks (OSNs), fake profiles, and general definitions related to Machine Learning .

In the next chapter we'll discuss some recent related work concerning fake profile detection.

Chapter 3

Fake profile detection : State of the art

3.1 Introduction

In this section on "Fake profile detection : State of the art", we present a literature review focused on the detection mechanisms for fake profiles. The widespread popularity of social networks has led to a rapid increase in the number of users accessing various platforms, resulting in a significant amount of data being shared and stolen on a large scale.

This study examines the literature on spam review detection, specifically employing the analysis of spammers' behavioral features. Numerous researchers have developed various methods to identify fake accounts, and this study aims to evaluate their contributions by comparing them to previous research.

A range of methods for detecting counterfeit profiles revolve around analyzing social network profiles and examining attributes or patterns that help distinguish genuine and fraudulent accounts. Various elements from profiles and posts are extracted, and algorithms are utilized to establish a classification framework for detecting fake accounts.

3.2 Selected articles

3.2.1 A novel machine learning-based framework for detecting fake Instagram profiles 2022

Keshav Kaushik, Akashdeep Bhardwaj, Manoj Kumar, Sachin Kumar Gupta, and Abhishek Gupta conducted a study to find an effective method for identifying fraudulent accounts and automatically generated spam accounts on Instagram. To detect the dataset used in the implementation, which represents both Instagram fake spammer and genuine accounts, two algorithms were employed. Prior to presenting the final precision and accuracy results, they conducted performance evaluation and trained a deep neural mode[21]

Instagram fake account detection model is depicted in the figure 3.1.

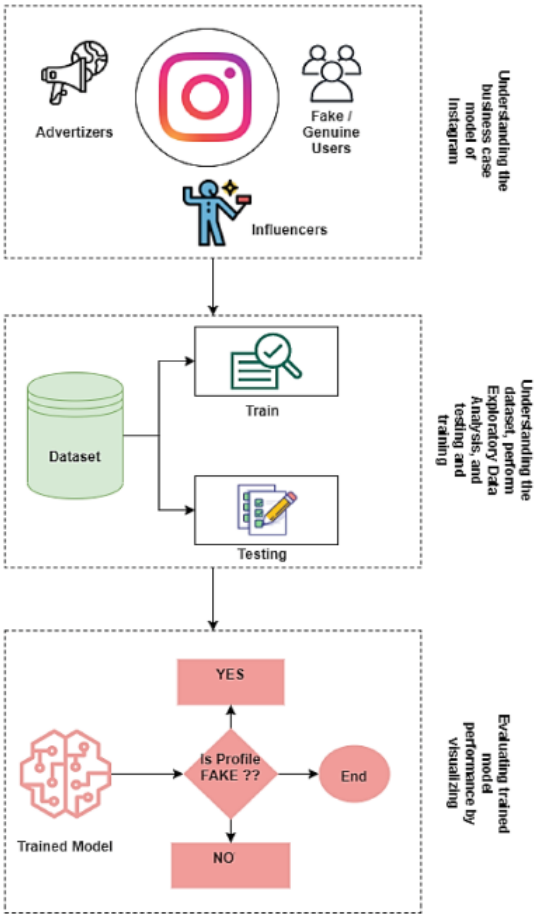


Figure 3.1: *Instagram fake account detection model [21]*

3.2.2 An Approach to Detect Fake Profiles in Social Networks Using Cellular Automata-Based PageRank Validation Model Involving Energy Transfer 2022

The text highlights the integration of PageRank algorithms with other methods to improve trustworthiness in these networks, including using deep Q-network architecture to identify social bots, detect cloned and fake profiles, and assess information credibility. It also discusses recent advancements in Social Network Analysis (SNA) and the application of graph-based techniques to understand user influence in social networks.[7]

Proposed flowchart of the system involving energy-based influence score is depicted in the figure 3.2.

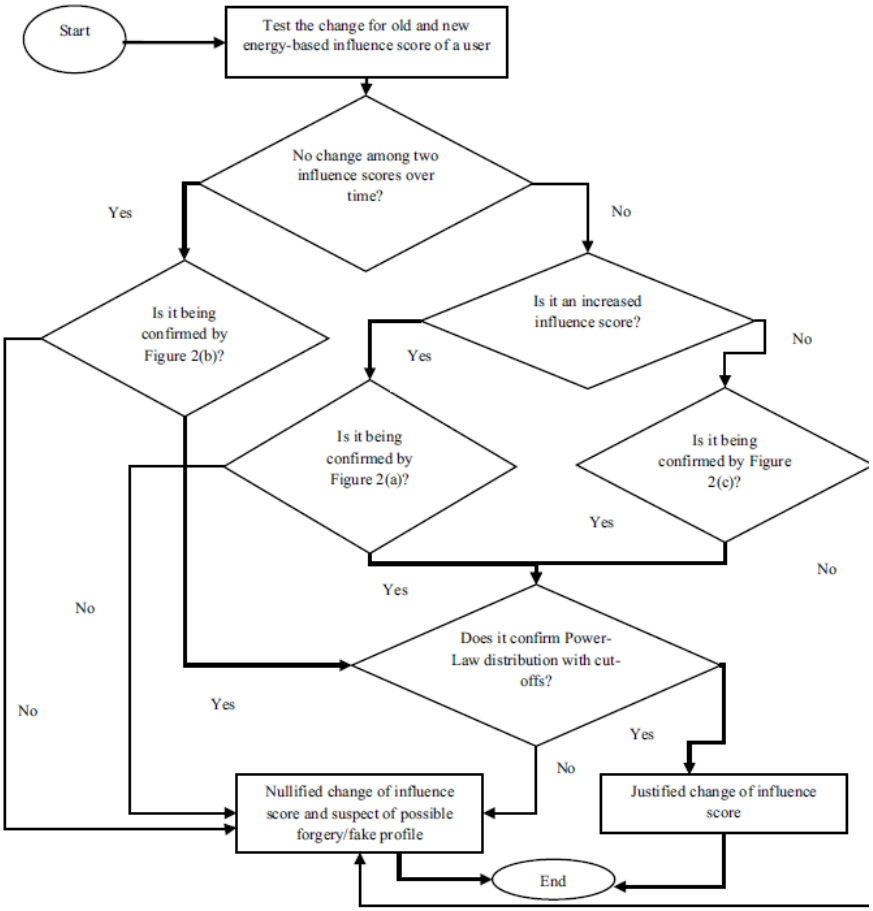


Figure 3.2: *Proposed flowchart of the system involving energy-based influence score [7]*

3.2.3 An across online social networks profile building approach: Application to suicidal ideation detection 2022

Atika Mbarek, Salma Jamoussi, and Abdelmajid Ben Hamadou developed a method to identify user profiles across multiple Online Social Networks (OSNs) and predict whether a user might be suicidal. They used supervised machine learning algorithms like RF, BN, SVM, DT, etc., with a real dataset created by Atika-Mbarek.

Their approach consists of three main steps:

Extracting matched profiles. Extracting features. Constructing a supervised model.

Each step employs its own machine learning algorithms to predict user suicidality. During testing, their strategy proved highly effective, especially when feature selection was applied during feature extraction. They achieved impressive precision, recall, and F1 scores of 88.9%, 85.7%, and 85.4%, respectively.[6]

Figure 3.3 shows Suicidal Profiles detection architecture.

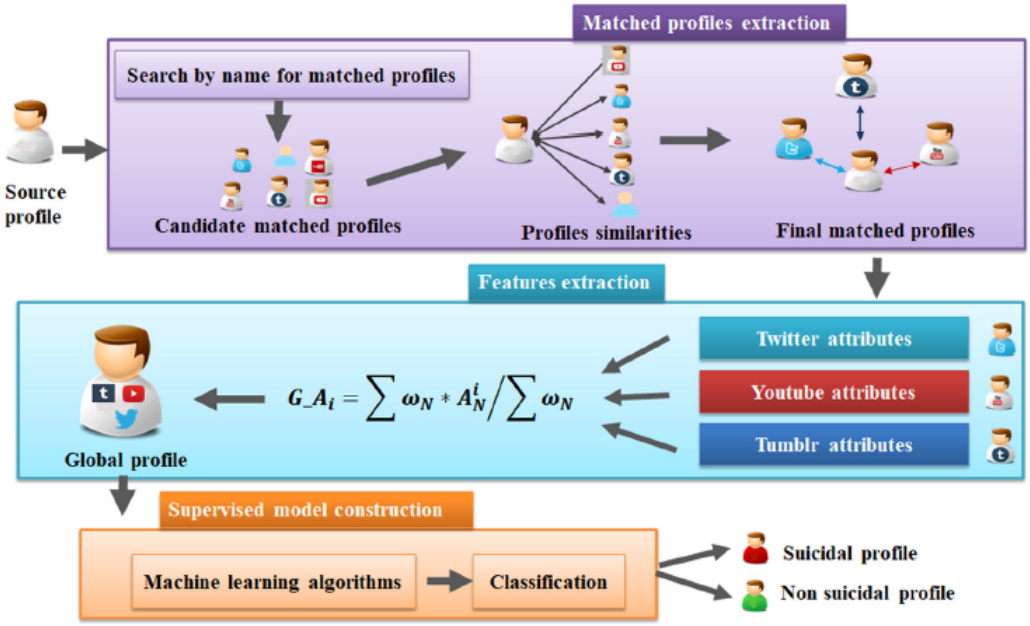


Figure 3.3: *Suicidal Profiles detection architecture.[6]*

3.2.4 Automatic Detection of Deaths from Social Networking Sites 2022

Nuhu Ibrahim(B) and Riza Batista-Navarro have developed an NLP-based method for detecting deaths. This method aims to facilitate the transfer of digital estates to family members or friends. They created a new corpus using data from Twitter and Wikidata. Various machine learning models, both traditional (RF, KNN, LR, and SVM) and deep learning (BiLSTM, CNN, and BERT), were trained on features extracted through different methods, including TF-IDF. The results demonstrated that the BERT model outperformed all other models, achieving a detection accuracy of 91.60 percent.[24]

The architecture of Steps involved in our methodology is as shown in Figure 3.4.

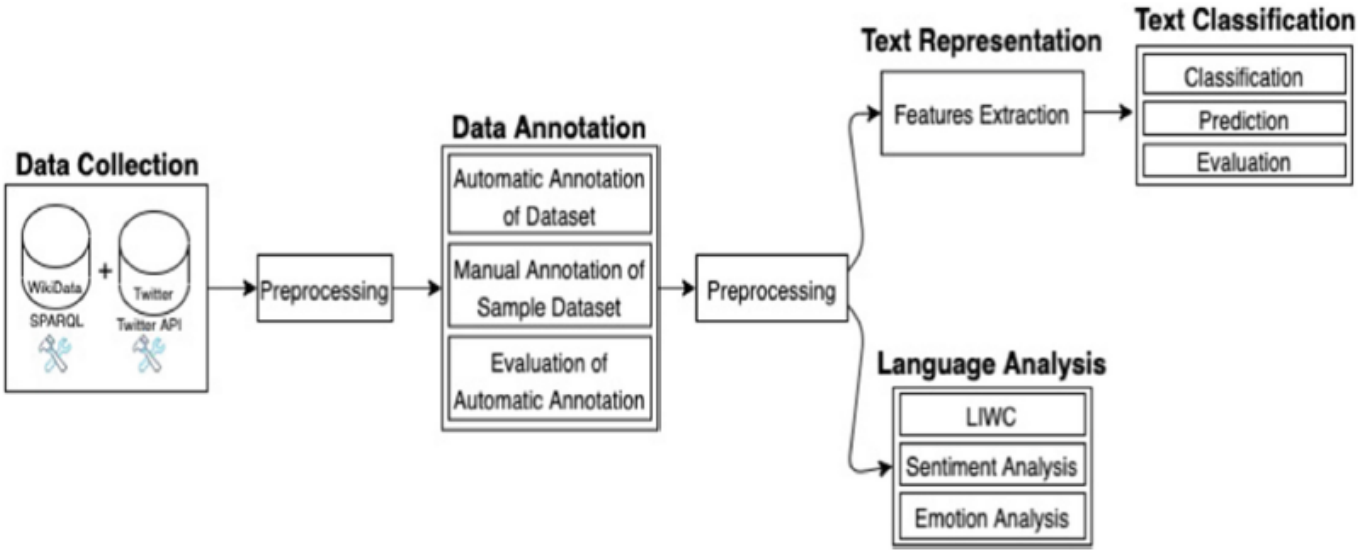


Figure 3.4: *Steps involved in our methodology[24]*

3.2.5 Deception detection on social media: A source-based perspective 2022

The study titled "Deception Detection on Social Media: A Source-Based Perspective," conducted by Khubaib Ahmed Qureshi, Rauf Ahmed Shams Malick, Muhammad Sabih, and Hocine Cherifi, focuses on detecting deception in social media content. While most current research emphasizes contextual and content-based methods to distinguish reliable sources, this work introduces a source-based approach.

This approach combines user attributes of social network users with the connectivity patterns of news spreaders. In a machine learning context, this hybrid technique leverages user profile data and network metrics from the community sharing the news.

Among 14 categories, the top three classifiers (XG Boost, Random Forest, and Decision Tree) were identified using datasets from Politifact and GossipCop. Notably, the "XG Boost" model outperforms its competitors, achieving an accuracy of 92% on Politifact and 91% on GossipCop based on the results.[30]

the following figure 3.5 Complete processing framework.

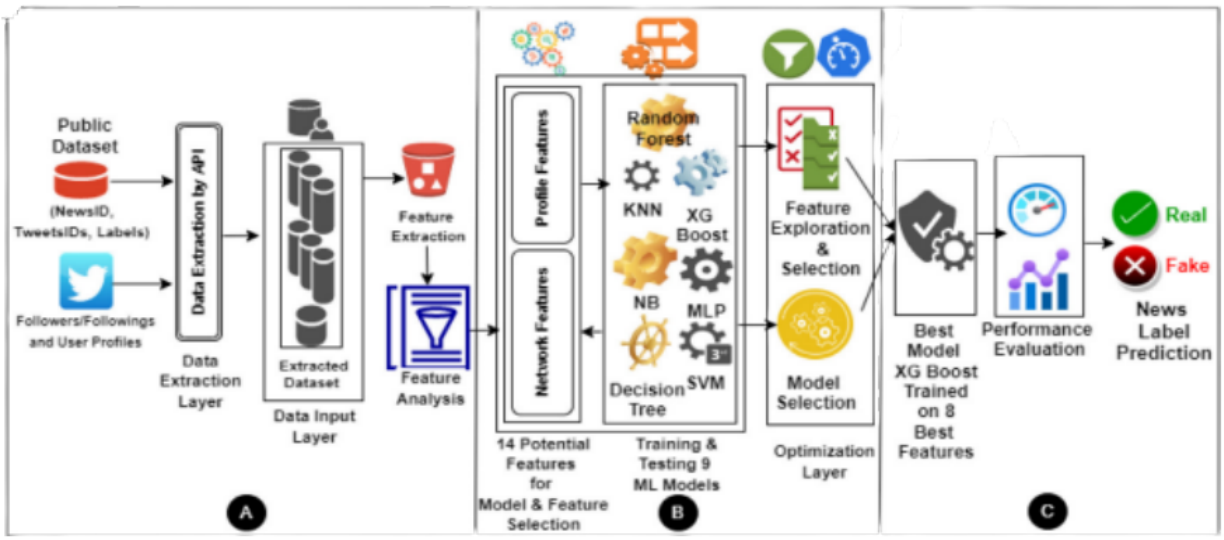


Figure 3.5: *Complete processing framework.[30]*

3.2.6 Detection and Classification of Genuine User Profile Based on Machine Learning Techniques 2022

In this paper, Prathyakshini, Nikitha Saurabh, Pratheeksha Hegde, and Preethi Salian introduced a classification model to distinguish between genuine and non-genuine users on social media platforms such as Instagram, Facebook, and Twitter. They conducted a comparison of various classification algorithms, including SVM, Neural Network, and Random Forest, using datasets from Kaggle. The results indicated that Random Forest outperformed the others, achieving an impressive accuracy rate of 95%. Consequently, Random Forest was selected as the primary classifier for their proposed model.[43]

The System Design is shown in the figure 3.6.

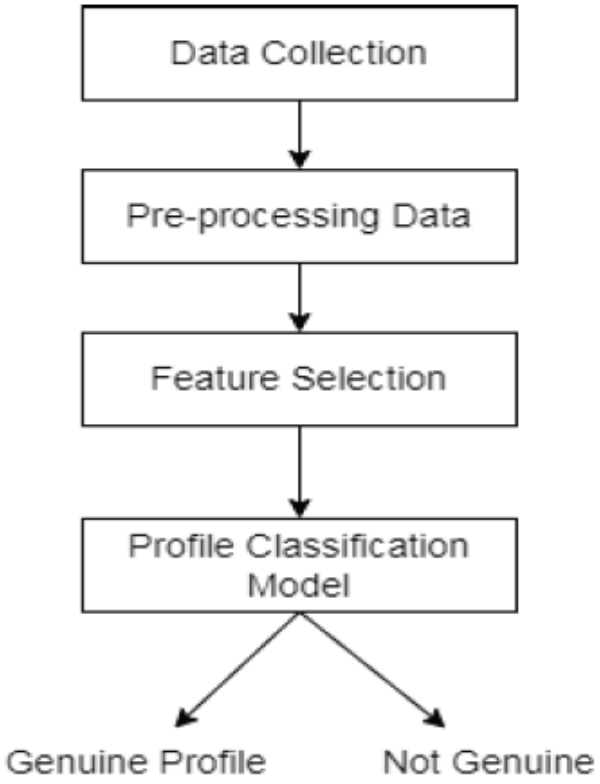


Figure 3.6: *System Design [43]*

3.2.7 Detection of Fake and Clone Accounts in Twitter Using Classification and Distance Measure Algorithms 2022

In a paper titled "Detection of Fake and Clone Accounts in Twitter Using Classification and Distance Measure Algorithms" authored by S. Siva Rama Krishna, K. Umakanth Reddy, T. Anji Reddy, A. Saiteja, and R. Sumanjali, machine learning supervised algorithms were applied to Twitter data. Their proposed algorithm consists of six modules: Text Cleaning, Bag of Words (BOW) Model, Training and Testing Datasets, Feature Scaling, KNN Algorithm, Prediction, and Experimental Results.

The recommended method, utilizing KNN and conducting a comparative analysis with NB to detect fake accounts, achieved an accuracy of 70.5% for KNN and 65% for NB.[47]

Figure 3.7 depicts the overall system architecture for detecting fake user accounts.

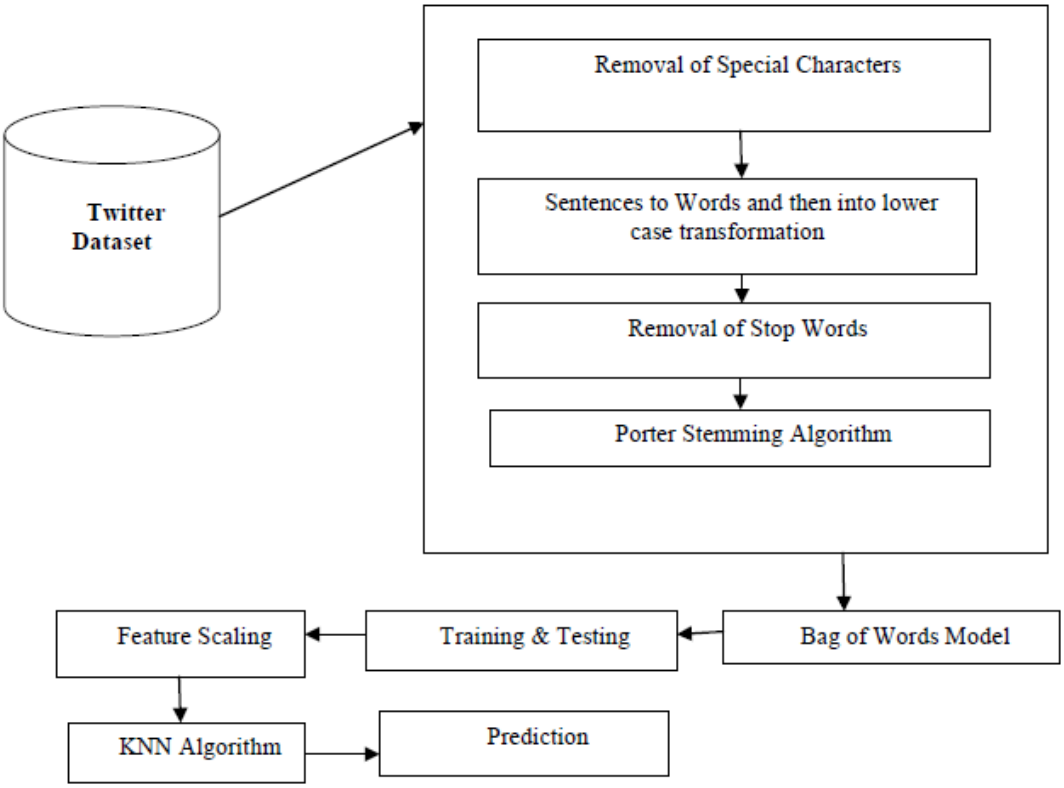


Figure 3.7: *Depicts the overall system architecture for detecting fake user accounts [47]*

3.2.8 Detection of Fake and Clone accounts in Twitter using Classification and Distance Measure Algorithmss 2020

Sahil Mulani, Vani Deshpande, Sana Inamdar, and Rohit Satavekar conducted a research study titled "Detection of Fake and Clone Accounts in Twitter using Classification and Distance Measure Algorithms." They employed a rule-based approach to differentiate between fake and genuine profiles with the aim of identifying fake profiles. Their research utilized Distance Measure Algorithms on datasets collected from MIB initiatives.

The rule set they applied achieved a commendable performance, with an accuracy rate of 90.2% in distinguishing between real and fake accounts.[48]

Figure 3.8 shows the architecture of the proposed system.

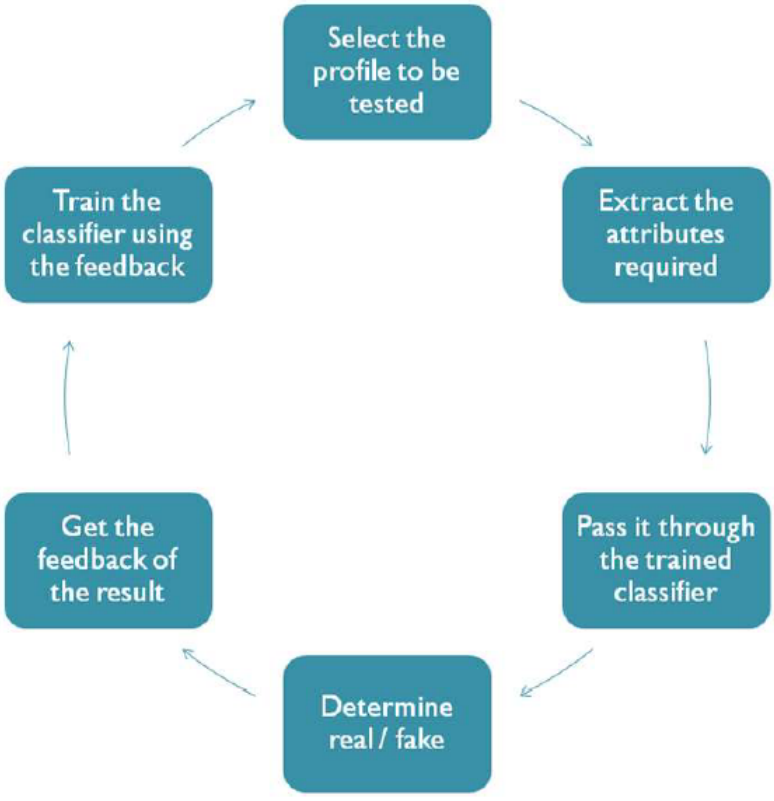
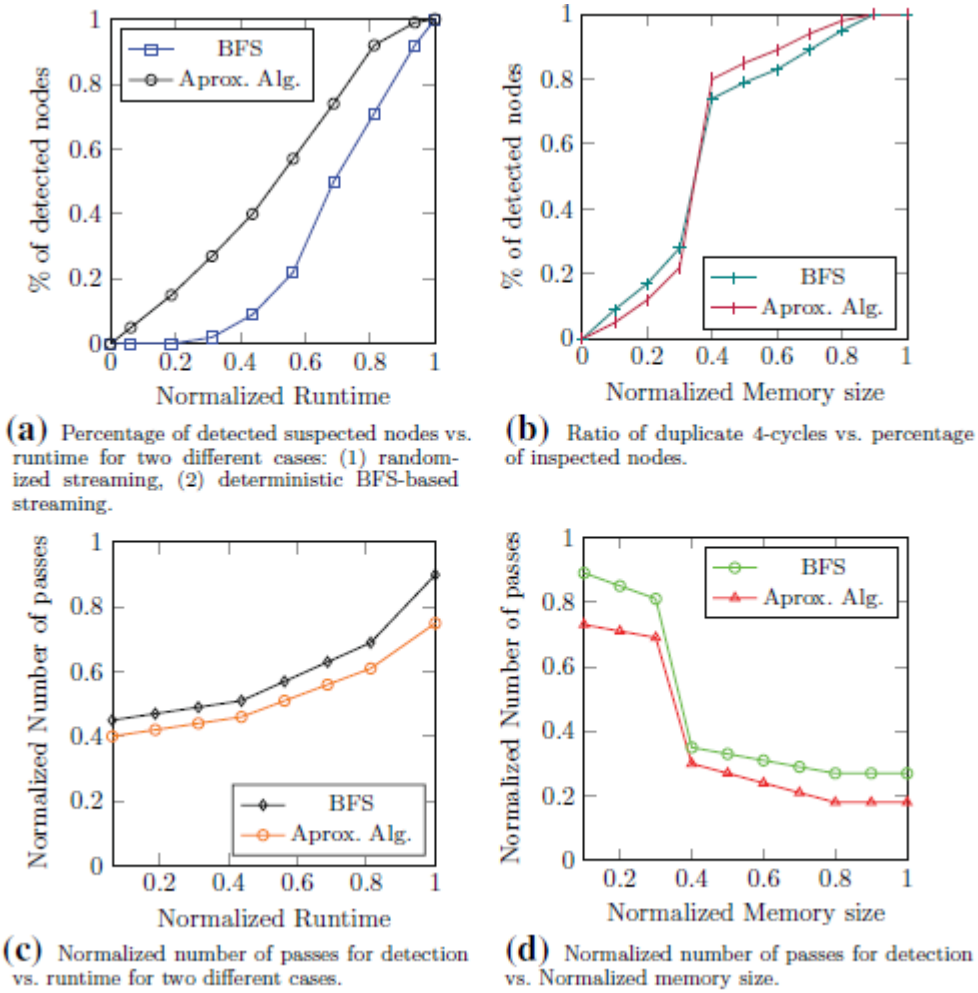


Figure 3.8: *Architecture of proposed system[48]*

3.2.9 Detection of fickle trolls in large-scale online social networks 2022

Hossein Shafei and Aresh Dadlani developed a paper titled "Detection of Fake Trolls in Large-Scale Online Social Networks." The paper explores detection methods at three different scales: Single Machine, Streaming, and Massively Parallel. They compare these methods to the two-hop neighbor discovery process and employ the Depth-First-Search (DFS) algorithm to identify suspicious nodes. Additionally, they use the Breadth-First-Search (BFS) algorithm to determine the order in which edges are processed by the machine.[51]

The figure 3.9 shows the evaluation of the streaming approach



(a) Percentage of detected suspected nodes vs. runtime for two different cases: (1) randomized streaming, (2) deterministic BFS-based streaming.

(b) Ratio of duplicate 4-cycles vs. percentage of inspected nodes.

(c) Normalized number of passes for detection vs. runtime for two different cases.

(d) Normalized number of passes for detection vs. Normalized memory size.

Figure 3.9: *Evaluation of the streaming approach[51]*

3.2.10 Fake accounts detection system based on bidirectional gated recurrent unit neural network 2022

Faouzia Benabbou, Hanane Boukhouima, and Nawal Sael have proposed a system for detecting fake accounts in social networks, specifically using a BiGRU deep learning model. They applied this system to Twitter, but it can be adapted for other social networks as well. The research utilized a balanced dataset generously shared by Cresci et al., which consisted only of comments. They employed word embedding techniques to maintain the context and syntax of comments. For each account, the content of tweets was consolidated into a single document and transformed into a vector space using GloVe, as illustrated in Figure 3.10.

The results revealed that the BiGRU approach outperformed other models in all aspects, achieving an impressive accuracy rate of 99.44%.[16]

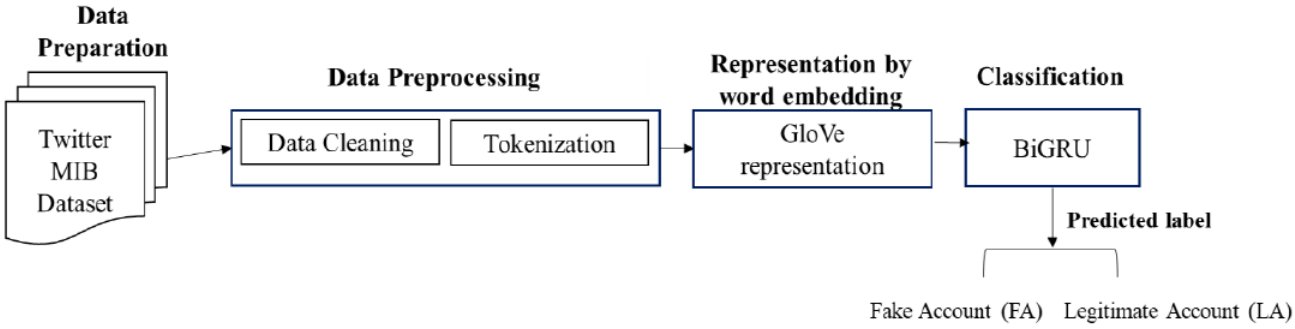


Figure 3.10: *The proposed method for fake account detection[16]*

3.2.11 Fake profile recognition using big data analytics in social media platforms 2022

"Hafiz Muhammad Faisal Shehzad, Mazhar Javed Awan, Muhammad Asad Khan, Zain Khalid Ansari, and Awais Yasin" have developed a project focused on recognizing fake profiles using big data analytics on social media platforms. They employed various algorithms, including SVM (with an accuracy of 88.8%), MLPC (with an accuracy of 92%), and KNN (with an accuracy of 90%), to enhance accuracy in datasets CK, Oulu, and MMI.

In their project, they utilized the Random Forest algorithm, achieving an impressive accuracy of 94%. The study involved data acquisition, feature engineering, and training data using a random forest classifier. They utilized several libraries for their research, including Pandas, Numpy, Sklearn, and Matplotlib, among others.[35]

The proposed model diagram is shown in Figure 3.11

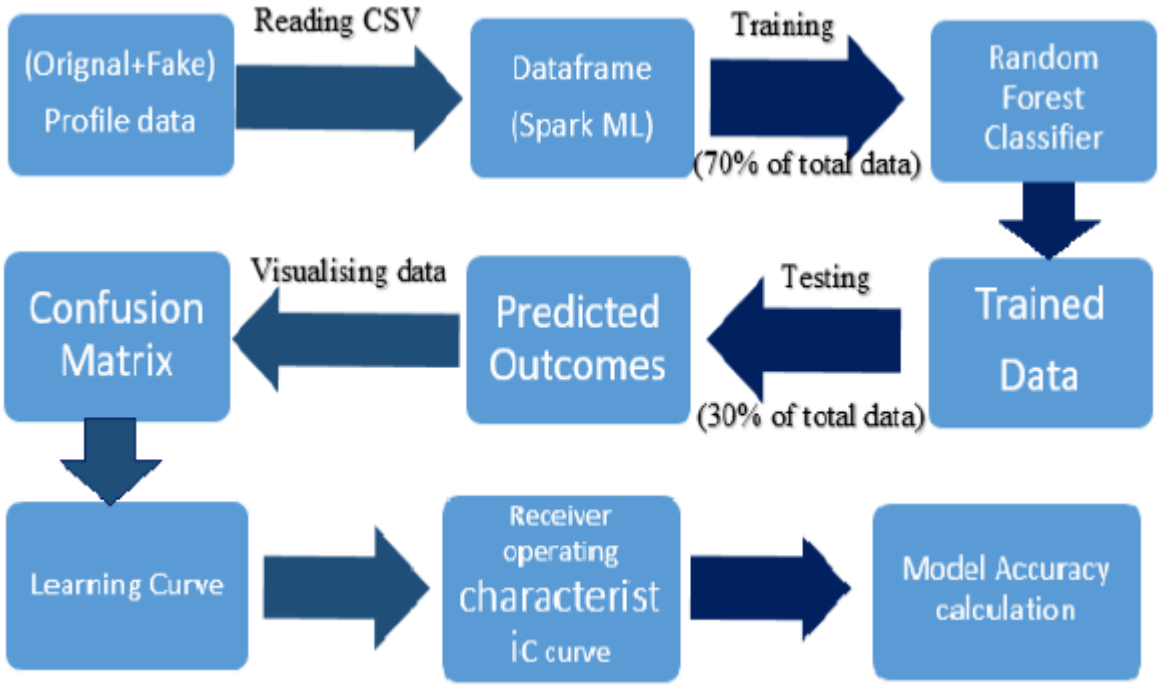


Figure 3.11: *Proposed model diagrams[35]*

3.2.12 Feature Selection for Identification of Fake Profiles on Facebook 2022

John Benyen Munga and Prabu Mohandas conducted research focused on identifying effective features for the detection of fraudulent accounts, particularly those with limited profile information. They collected features from a dataset comprising 1337 fake profiles and 1481 genuine profiles.

To select the most relevant features, they applied entropy and information gain techniques. Out of the initial 25 attributes, the proposed approach identified only eight as effective for detecting fake accounts, with five of them selected based on information gain. They demonstrated that using this minimal set of features with higher information gain, and through two rounds of experimentation with different algorithms (RF, SVM, and NN), they achieved improved accuracy in fraudulent account detection.[38]

Figure 3.12 shows the steps involved before the data preprocessing in two rounds.

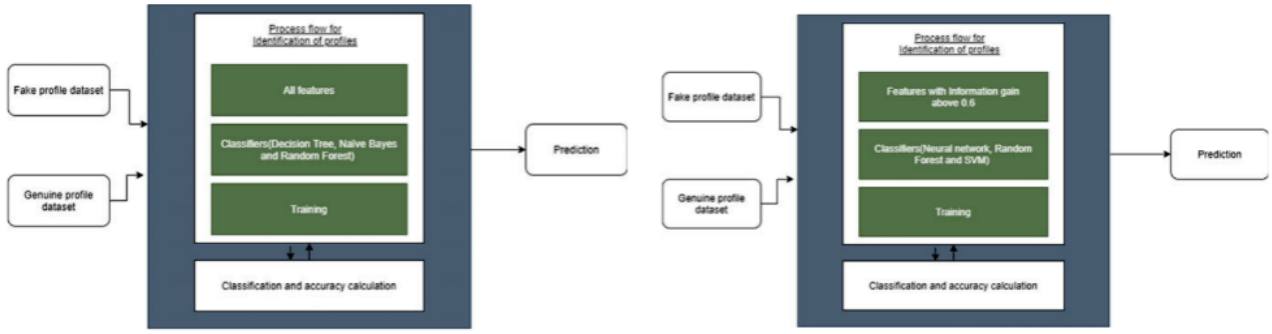


Figure 3.12: *The steps involved before the data preprocessing in two rounds [38]*

3.2.13 Is it Sarrah Rahamah? A supervised classification model to detect fake identities on Facebook within the Sudanese community 2022

The paper titled "Is it Sarrah Rahamah? A Supervised Classification Model to Detect Fake Identities on Facebook within the Sudanese Community" was developed by Mariam Elhussein. The research focuses on the issue of fake accounts with fake identities in the Facebook group for Sudanese people. While human phony accounts are often overlooked, this study concentrates on automatic and semiautomatic accounts within their cultural context.

The study interviewed 250 Sudanese Facebook users who had encountered eight of these fake identities. Data from both confirmed fake and legitimate accounts were manually collected. The dataset, which included 231 instances, was imbalanced, but this was addressed by using SMOTE oversampling.

The supervised classification systems (DT, LR, RF, SVM, and NB) achieved an AUC of 0.96% and an accuracy rate of up to 89.7%.[14]

See the figure 3.13 The results of machine learning classifers with and without oversampling.

	With over-sampling		Without over-sampling	
	Accuracy	AUC	Accuracy	AUC
Decision tree	89.26%	0.93	97.84%	0.50
Random forest	89.71%	0.96	97.84%	0.62
Naïve-Bayes	85.22%	0.93	89.73%	0.59
SVM	87.47%	0.94	96.56%	0.61
LR	86.56%	0.95	97.84%	0.59

Figure 3.13: *The results of machine learning classifers with and without over-sampling [14]*

3.2.14 Profiling Fake News: Learning the Semantics and Characterisation of Misinformation 2022

"The Semantics and Characterization of Misinformation" is a research work developed by Swati Agarwal and Adithya Samavedhi. In this paper, they focus on profiling false news and identifying the characteristics that distinguish it from real news. To validate their set of characteristics, they utilize four commonly used open-source datasets.

They evaluate the effectiveness and significance of their suggested feature set using various artificial, recurrent neural network, and machine learning models, including Naive Bayes (GNB), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), XGBoost (XGB), and Support Vector Machine (SVM). These models achieved an impressive accuracy of 90

Their findings indicate that ensemble-based classifiers outperform individual classifier models. Additionally, the FNM and ISOT datasets demonstrate better performance compared to the KE and Liar datasets.[1]

Figure 3.14 shows the Performance results of classifiers employed on all experimental datasets.

	FN Master			ISOT			Liar-Liar			KE		
	P	R	A	P	R	A	P	R	A	P	R	A
GNB	0.78	0.87	0.83	0.75	0.59	0.63	0.49	0.50	0.54	0.67	0.48	0.55
RF	0.98	0.96	0.98	0.92	0.92	0.92	0.56	0.56	0.58	0.63	0.49	0.50
DT	0.95	0.95	0.96	0.85	0.85	0.86	0.52	0.52	0.53	0.52	0.51	0.50
KNN	0.90	0.88	0.92	0.89	0.88	0.88	0.55	0.54	0.57	0.61	0.56	0.59
XGB	0.97	0.96	0.98	0.92	0.92	0.92	0.56	0.55	0.58	0.57	0.55	0.55
SVM	0.94	0.93	0.96	0.93	0.93	0.93	0.56	0.54	0.57	0.52	0.50	0.50
ANN-1	0.94	0.92	0.95	0.92	0.91	0.92	0.28	0.50	0.56	0.36	0.40	0.45
ANN-2	0.95	0.91	0.95	0.92	0.92	0.92	0.56	0.53	0.57	0.58	0.52	0.55
ANN-3	0.90	0.92	0.94	0.89	0.88	0.89	0.57	0.52	0.56	0.20	0.33	0.27
ANN-4	0.93	0.94	0.95	0.90	0.89	0.89	0.57	0.51	0.56	0.25	0.29	0.32
LSTM-1	0.56	0.52	0.74	0.96	0.96	0.96	0.59	0.57	0.60	0.23	0.28	0.32
LSTM-2	0.55	0.52	0.75	0.96	0.96	0.96	0.58	0.58	0.57	0.14	0.33	0.41
LSTM-3	0.38	0.50	0.77	0.27	0.50	0.55	0.29	0.50	0.57	0.28	0.33	0.36
RNN-1	0.38	0.50	0.77	0.47	0.48	0.51	0.55	0.55	0.57	0.20	0.25	0.27
RNN-2	0.59	0.50	0.77	0.75	0.57	0.61	0.54	0.54	0.56	0.23	0.31	0.36
RNN-3	0.38	0.50	0.77	0.27	0.50	0.55	0.54	0.53	0.57	0.32	0.42	0.50
GRU-1	0.54	0.52	0.74	0.94	0.93	0.93	0.57	0.57	0.54	0.14	0.33	0.41
GRU-2	0.54	0.51	0.75	0.94	0.93	0.93	0.58	0.50	0.57	0.14	0.33	0.41
GRU-3	0.38	0.50	0.77	0.27	0.50	0.55	0.29	0.50	0.57	0.28	0.33	0.36
BiLSTM-1	0.85	0.71	0.85	0.99	0.99	0.99	0.57	0.57	0.58	0.12	0.33	0.36
BiLSTM-2	0.67	0.72	0.73	0.99	0.99	0.99	0.59	0.59	0.60	0.30	0.34	0.41
BiLSTM-3	0.38	0.50	0.77	0.80	0.60	0.64	0.29	0.50	0.57	0.48	0.38	0.45

Figure 3.14: *Performance results of classifiers employed on all experimental datasets[1]*

3.2.15 RunMax: fake profile classification using novel nonlinear activation in CNN 2022

The article "RunMax" by Putra Wanda introduces a new approach to detect fake profiles using the RunMax nonlinear activation function in a Convolutional Neural Network (CNN). They collected and analyzed profile data from over 3000 unique users to build their dataset. The results showed that RunMax achieved better accuracy in identifying fake profiles (83.34% test accuracy and 81.72% training accuracy) compared to the traditional SoftMax activation function and standard linear activations.[58]

Figure 3.15 shows a Comparison among classifiers with the CNN architecture.

Figure 3.16 shows a Comparison of precision, recall, and F1-Score among algorithms.

Activation type	Training accuracy (%)	Testing accuracy (%)
CNN + No activation	80,51	79,54
CNN + SoftMax	82,31	81,33
CNN + RunMax	83,34	81,72

Figure 3.15: *Comparison among classifiers with the CNN architecture [58]*

Algorithm	Precision (%)	Recall (%)	F1-Score (%)
Naïve bayes	86.91	86.95	87.02
GB (n estim = 50)	90.65	90.69	91.01
LR	90.48	90.58	90.60
SVM (rs = 31.6)	90.04	87.34	87.24
Proposed CNN	94.01	93.22	93.41

Figure 3.16: *Comparison of precision, recall and F1-Score among algorithms. [58]*

3.2.16 Social Media Fake Profile Detection Using Data Mining Technique 2022

The study conducted by Nitika Kadam and Sanjeev Kumar Sharma aims to investigate techniques for identifying false profiles on various social media platforms, with a focus on machine learning and data mining approaches. They also reviewed existing methods in this context.

Their paper proposes a data mining approach using a dataset available on GitHub. The dataset was refined with expert assistance, and popular data mining algorithms were applied, including KNN, SVM, ANN, Bayesian, and C4.5 decision trees. Two validation ratios, 70-30% and 80-20%, were employed to evaluate the performance of these techniques, as shown in figure 3.17.

The results indicate that a 70% to 30% ratio is effective for accuracy and error rate, while an 80% to 20% ratio is effective for resource consumption. These findings led to the development of two accurate and effective classification techniques for fake profile detection, contributing to

the advancement of a more sophisticated model for detecting fake profiles.[27]

Table 3.17 shows The efficiency of classification outcomes for the test datasets.

Algorithms	Performance Summary for 70-30%				Validation Summary for 80-20%	
	Accuracy	Error rate	Memory	Time	Accuracy	Error rate
C4.5	86.5%	13.5%	14029 KB	267 MS	83.4%	16.6%
Bays	84.3%	15.7%	13898 KB	289 MS	82.9%	17.1%
ANN	97.4%	2.6%	15294 KB	365 MS	95.7%	4.3%
SVM	96.5%	3.5%	15164 KB	376 MS	94.2%	5.8%
KNN	84.2%	15.8%	13772 KB	398 MS	83.5%	16.5%

Figure 3.17: *The efficiency of classification outcomes for the test datasets.[27]*

3.2.17 Using Social Media to Detect Fake News Information Related to Product Marketing: The FakeAds Corpus 2022.

The paper "Using Social Media to Detect Fake News Information Related to Product Marketing" was authored by Noha Alnazzawi, Najlaa Alsaedi, Fahad Alharbi, and Najla Alaswad. The main objective of this study was to create a dataset for the research community to use in Twitter fake news detection. Specifically, the dataset focuses on identifying misleading content that falsely promotes products. The study utilized machine learning (ML) algorithms in conjunction with the "FakeAds" corpus and achieved impressive results, as indicated by an F-score of 0.815 in their annotation efforts.[57]

The figures 3.18 shows the distribution of fake and real tweets in the FakeAds corpus. The figures 3.19 shows the The distribution of product types in FakeAds corpus.[57]

Figure 3.18: *The distribution of fake and real tweets in the FakeAds corpus.[57]*

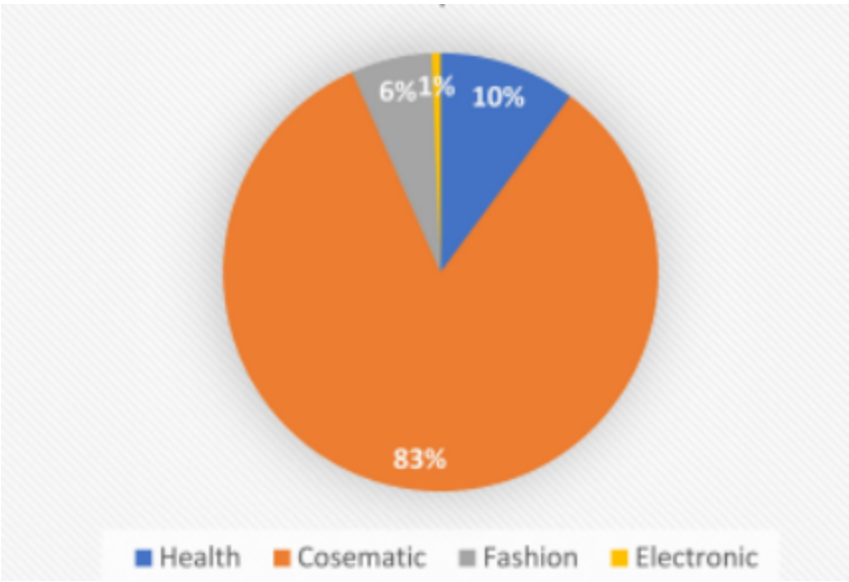


Figure 3.19: *The distribution of product types in FakeAds corpus.[57]*

3.2.18 Social Media Identity Deception Detection: A Survey 2022

Social media platforms have become increasingly prevalent sources for identity deception. In recent years, numerous cases of identity deception on social media have come to light.

A comprehensive analysis of identity deception attacks on social media platforms has been carried out, encompassing various forms of deception, including fake profiles, identity theft, and identity cloning.

The research challenges in this domain can be broadly categorized into general challenges that are applicable across all identity deception detection techniques and specific challenges that pertain to particular subcategories of identity deception detection methods. These challenges represent areas where further research and innovation are needed to effectively combat identity deception on social media platforms.[2]

The figure 3.20 shows Social media deception

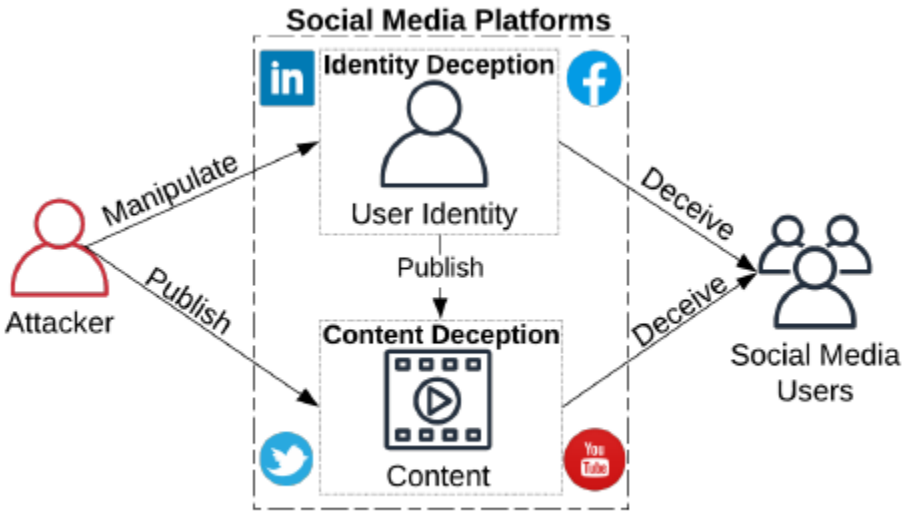


Figure 3.20: *Social media deception[2]*

3.2.19 Spammer Detection Approaches in Online Social Network (OSNs): A Survey 2022

In a paper presented at the International Conference on Applied Cryptography and Network Security, the authors categorized the spam detection framework into different segments, which included simultaneous identification and removal of spam content. They observed that spammers often disseminate spam through hashtags, URLs, and spammy text in messages. They also explored a method that clusters related tweets based on their text content and shortened URLs.[54]

The figure 3.21 shows Social media deception

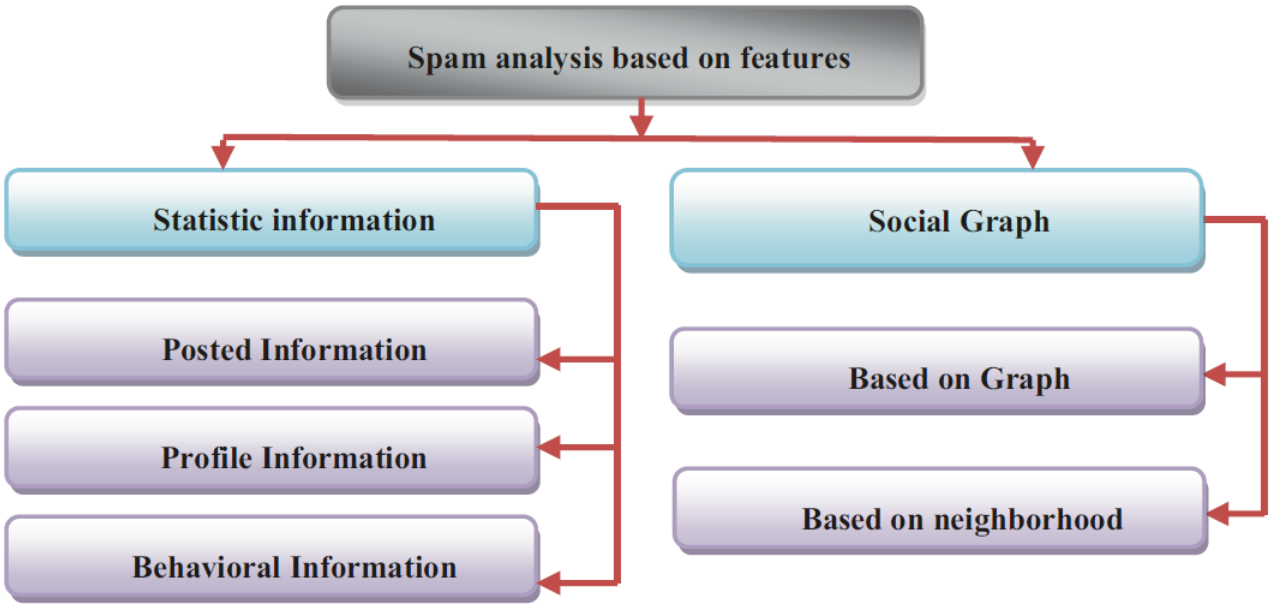


Figure 3.21: *Various categories of features[54]*

3.2.20 Fake Profiles Identification on Social Networks With Bio Inspired Algorithm 2022

Nadir Mahammed, Souad Bennabi, Mahmoud Fahsi, Badia Klouche, Nadia Elouali, and Chourouk Bouhadra have developed an approach for detecting fake profiles on social media. This approach combines a machine learning algorithm with a bio-inspired algorithm. They used a dataset from the Facebook social network to identify fake profiles.

Comparatively, their bio-inspired technique outperformed other supervised classification models, achieving an impressive accuracy rate of 98.9%. This suggests that their approach is more effective in identifying fake profiles than other algorithms.[40]

Figure 3.22 shows SBO+k-means flowchart.

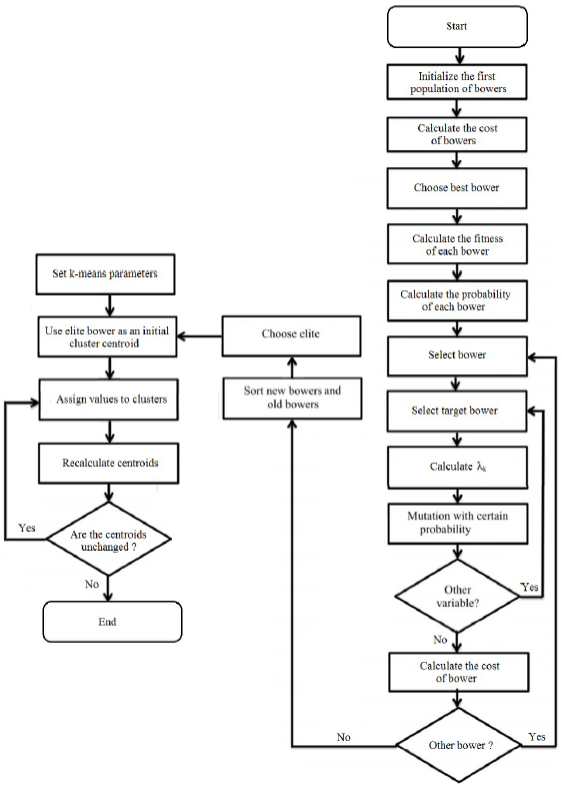


Figure 3.22: *SBO+k-means flowchart[40]*

3.2.21 Effective Spam Bot Detection Using Glow Worm-Based Generalized Regression Neural Network 2022

In this study, conducted by A. Praveena and S. Smys, the focus was on detecting spam accounts using a combination of machine learning and an optimization process. The optimization method employed in this research is known as "glow worm optimization," which is used to select optimal features. This helps reduce computational time and improves the accuracy of the classifier. For classification, they used the generalized regression neural network.

The researchers tested their methodology using the Social Honeypot dataset, which includes data on 19,276 original users and 22,223 spam users. These users generated a significant volume of tweets.

The results indicated that the proposed GWO-GRNN approach outperformed the deep Q-learning method, achieving an accuracy rate of 95%. This suggests that GWO-GRNN is the most effective strategy in terms of processing and performance compared to the current technique.[44]

Figure 3.23 shows GWO-GRNN flow chart for spam bot detection.

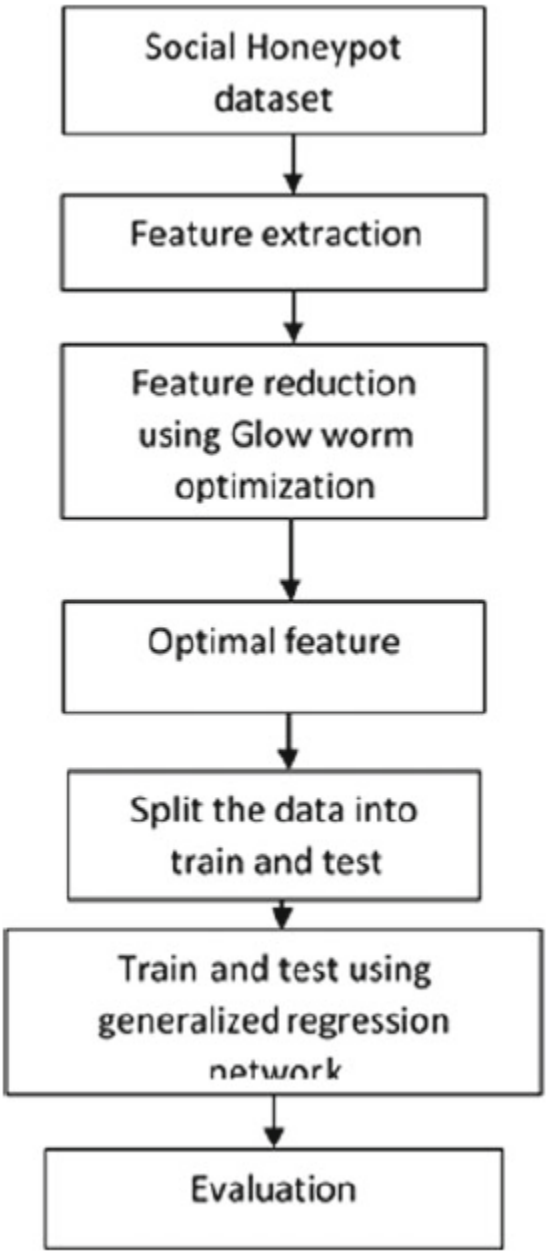


Figure 3.23: *GWO-GRNN flow chart for spam bot detection.[44]*

3.3 Synthesize and discussion

We have summarized the literature presented in Sect. 3 in tabular format Tables 3.1 and 3.2 and given a rebuttal of what people have researched in their works between 2020 and 2022 and also we have stated the findings and limitations of the reviewed papers.

Reference	OSN	ML Type	ML algorithme	Dataset	Resultat	Metaheuristique
[21]	Instagram	supervised	NN, SVM, LR, NB	120 576	Acc=95%	/
[7]	YouTube LinkedIn	unsupervised	PageRank ECAs	380000 2500 514	/	/
[6]	Twitter YouTube Tumblr	supervised	SVM, RF,DT BN , adab,XGBoost	111	Acc(XGBoos) =88% ACC(SVM)= 91,5%	/
[24]	twitter	supervised	RF, KNN, LR, SVM, BiLSTM, CNN ,BERT	/	Acc(BERT) =91,60%	/
[30]	twitter	supervised	RF, DT, MLP, KNN, SVM,XG,Boost	28167 124	ACC=92% ACC=91%	/
[43]	Facebook Instagram Twitter	supervised	RF,SVM,NN	698 2820 11420	ACC(RF)=9 5%	/
[47]	twitter	supervised	Distance measure algorithme (KNN),NB	1491	Acc(KNN)= 90.2% Acc(NB)= 65.5%	/
[51]	twitter	Search algorithm	/	37556, 378296, 365604, 324513, 331822, 388453, 368886, 397229, 381013		DFS , BFS
[44]	Twitter	/	/	19276 22223	ACC=95%	GWO GRNN
[16]	Twitter	supervised	BiGRU Glove	2818	Acc(BiGRU Glove)=99. 44%	
[35]	Facebook Twitter	supervised	SVM, MLPC ,KNN ,RF ,NB	51 ,28001 ,6194	Acc(RF)= 94%	/
[40]	Facebook	unsupervised	Kmeans	1244	Acc(SBO+ Kmeans) =98.9%	SBO
[38]	Facebook	supervised	RF,SVM, NN	/	Acc(Round 01): RF=96.45% ,SVM=94.8 5% ,NN=48.49 %. Acc(Round	/

Table 3.1: Comparative study of studied research - part 01

[38] suite					02): RF=99.46% ,SVM=99.94% ,NN=94.13 %	
[14]	Facebook	supervised	LR,SVM,DT,NB, RF	231	DT=0,89 RRFM=0,89 SVM=0,87 NB=0,85 LR=0,86	/
[1]	Twitter Facebook Instagram	supervised	GNB,RF,DT,KNN, XGB, SVM	23504 17949 11509	ACC(SVM)= 90%	/
[58]			CNN FNAL+RUNMAX	3000	Acc(Training) =83.34% Acc(Test) =81.72%	/
[27]	Twitter	supervised	C4.5 , SVM ,ANN BAYS ,KNN	2820	Acc(70;80): C4.5=86.5 % Bayes=84.3 % ANN=97.4 % SVM=96.5 % KNN=84.2 % Acc(80;20) C4.5=83.4 % Bayes=82.9 % ANN=95.7 % SVM=94.2 % KNN=83.5 %	/
[2]	Twitter Facebook	supervised	SVM,RF,LSTM,SMOTE,ENN,GLOVE,NN			/
[54]	Twitter Facebook	supervised	RF,Feauture selection ,pso	145000	ACC=97,17 % ACC(RF)=72,3% ACC(SVM)= 93,67%	ACC(PSO)=91,20%
[57]	twitter	supervised	ML Algorithm	5000	Fscore=0.815	/

Table 3.2: Comparative study of studied research - part 02

3.4 Discussion

This discussion analyzes a table of results from various research articles to identify patterns, trends, and noteworthy findings. This comparative analysis offers valuable insights into the

research topic, enhancing our understanding of the overall research landscape and potential avenues for further exploration.

- **The interference of twitter in OSN research:**

  Twitter is widely acknowledged in academic research due to its frequent mention in scholarly articles. Its large user base, real-time nature, and data availability make it a valuable resource for studying online social networks. Its prevalence in academic research highlights its importance in understanding social phenomena and user behavior.

- **Detection of Fake Profiles:**

  Detecting fake profiles in OSN research is vital for social network integrity and security. Both supervised and unsupervised machine learning methods are used successfully to identify and categorize fake accounts, helping reduce fraud.

- **The superiority of Supervised Machine Learning:**

  Supervised ML, like RF,SVM and DT, is widely used for spotting fake Twitter profiles. They learn from labeled data, recognizing fake account patterns for accurate classification. Their prevalence in research suggests their superior performance over unsupervised methods.

- **Utilizing Large Datasets:**

  Research articles frequently stress the importance of employing substantial datasets, typically ranging from 100 to 20,000 instances, for evaluating fake profile detection algorithms. These diverse datasets enable comprehensive assessment of algorithm scalability, accuracy,

and limitations. Utilizing large datasets allows researchers to evaluate the robustness and generalizability of their detection models.

- **Exploring the Potential of Bio-Inspired Techniques:**

  While bio-inspired methods are not widely applied in OSN research, they show promise for achieving high accuracy in fake profile detection. Drawing inspiration from natural systems like neural networks, genetic algorithms, and swarm intelligence, these techniques emulate social dynamics and propagation processes in OSNs. Preliminary research into bio-inspired methods indicates their significant potential in improving accuracy by harnessing the inherent traits of social networks.

- **Synergistic Integration for Enhanced Accuracy:**

  Combining various detection methods, including supervised and bio-inspired approaches, enhances fake profile detection accuracy. Integrating supervised machine learning algorithms like RF and DT with bio-inspired techniques leverages their individual strengths. This synergy harnesses diverse methodologies and unique capabilities, resulting in improved accuracy, especially for complex and evolving fake profile detection challenges.

3.5 Conclusion

In this chapter, we've presented a theoretical foundation for comprehending OSNs technology. This groundwork enables us to address the issue of fake profiles and examine the latest related research aimed at addressing this security concern.

Summary points from this review are the following:

- Due to Twitter's default public information accessibility and easy access through Twitter APIs, it has become the primary target among OSN platforms.

- From the research studies, shows that the majority of both real and fake accounts typically have a profile picture and a name attribute.

- Supervised and unsupervised machine learning methods are both employed for fake profile detection and have proven effective in identifying fake accounts.

- The reviewed studies often utilize extensive datasets for fake profile detection testing, ensuring there's ample data that surpasses the algorithm's capacity, allowing it to recognize both profile attributes and fake profile characteristics.

Chapter 4

Our contribution

4.1 Introduction

The first phase of this study involved evaluating the performance of different machine learning algorithms using a dataset comprising profiles from Facebook, Twitter, and Instagram. Multiple performance indicators were considered to assess the effectiveness of the algorithms.

The scope of the study was defined based on its objectives and available resources. The primary goal was to analyze the performance of machine learning algorithms using specific datasets extracted from Facebook, Twitter, and Instagram profiles, and derive meaningful insights. The findings obtained from this analysis will be compared with existing research on the topic, leading to conclusions and insights.

4.2 Data processing

4.2.1 Dataset Collection :

Facebook dataset Description :

In the proposed work we get Facebook account dataset (xls file). The dataset was constructed from Facebook which is social media networking site. This dataset contain 1244 rows and 15

columns.

The dataset used in our research is the Facebook Dataset 9-9-2019 [41]; which contains 1244 accounts divided as follows:

- Real Accounts : It contains 1043 accounts, 100% human collected in a research project.
- Fake Accounts : It contains 201 fake accounts.

	Legitimate	Fake	Total
Records	1043	201	1244
Percentage	83.84 %	16.16%	100%

Table 4.1: Facebook dataset description

We extract following feature set from the collected dataset of Facebook. This feature set consists of 14 features which help to accurately classify the data.

- Name-Id.
- Link.
- Profile Picture.
- Number of Likes.
- Number of groups joined.

- Number of friends.
- Education status.
- Work(mentioned or not).
- Living place (mentioned or not).
- Relation-ship.
- CheckIn.
- Number of posts.
- Number of tags.
- profile intro.

Features Analysis : To separate genuine accounts and fake accounts on facebook, analysis of different Facebook characteristics in our framework are described below :

Feature number	Facebook feature	Why for
F1	Profile id	Unique profile ID provided to the user.
F2	Profile picture	Image of the profile holder is the real one or used some esteemed person's image like celebrity or politician.
F3	Likes	Used for appreciation for a particular topic in the form of posts or media content.
F4	Groups	The number of groups in which the user is joined.
F5	Mutual friends	The number of mutual friends.
F6	Education	Education mentioned or not.
F7	Work	Work exists or not.
F8	Living place	Living place mentioned or not.
F9	Relationship/family	Mentioned or not.
F10	Check In	Users who wish to announce their location to their friends on Facebook would tap a "check in" button to see a list of places nearby, and then choose the place that matches where they are.
F11	Posts	Content shared by the user in the form of text, audio and video in addition with hashtags and URLs.
F12	Tags	Facebook provides a tagging feature by which a user can tag a specific user (in comment or post). It's a process of pointing a specific user toward their posts.
F13	Intro	For identifying the account.

Table 4.2: Facebook dataset Features analysis

Twitter Dataset

We used a dataset that has been created using Twitter API in an other work. There are four main objects in Twitter API. These are : Tweets, Users, Entities and Places. Each of these objects have many attributes [15].

Tweets objects are the basic atomic building of all things. It has some attributes about general information of the tweets, e.g when the tweet was created, how many times the tweet has been liked, number of times a tweet has been retweeted etc. This attributes are not reachable

for the protected accounts [15].

Users objects can be anyone or anything. It has some attributes about general information of the accounts, e.g the number of tweets the user has liked, the number of followers the account has, the number of user the account is following etc. This attributes are also reachable for the protected accounts. We have selected our features mostly from this attributes because of its availability [15].

Entities objects provide metadata and additional contextual information about content posted on Twitter. It has some values, e.g. hashtags, media, urls in the tweet [15]. Places objects are named locations with corresponding geo coordinates [15]. The dataset contains 16 attributes.

	Legitimate	Fake	Total
Records	499	501	1000
Percentage	49.9 %	50.01%	100%

Table 4.3: Twitter dataset description

Attribute Name	Description of the attribute.
Description	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.
Favorites 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, allowing for Tweets issued by the user to be coauthored by another account.
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.

Table 4.4: Features of Twiter dataset

Instagram Dataset

The dataset has been taken from kaggle [1].It consists of two CSV files- train.csv (19 KB) and test.csv (4 KB) .The dependent variable, which is whether it is a fake or not fake account is categorical and it takes two values 0 (not fake) and 1 (fake) profile. The distribution of the training dataset is such that 50% is fake and the rest 50% is legitimate .[12]

[1]https://www.kaggle.com/free4ever1/instagram-fake-spammer-genuineaccounts

	Legitimate	Fake	Total
Records	348	348	696
Percentage	50%	50%	100%

Table 4.5: Instagram dataset description

Attribute Name	Description of the attribute
Profile pic	User has profile picture or not
Nums/length username	Ratio of number of numerical chars in username to its length
Fullname words	full name in word tokens
Nums/length fullname	Ratio of number of numerical characters in full name to its length
Username	Are username and full name literally the same
Description length	Bio length in characters
External URL	Has external URL or not
Private	Private or not
Posts	Number of posts
Followers	Number of followers
Follows	Number of follows

Table 4.6: Features of Instagram dataset

4.2.2 Dataset preprocessing

Extensive data has been collected from various sources, such as the internet, questionnaires, and tests. However, these datasets often suffer from distortions, noise, and missing values. Data preprocessing is a vital set of techniques that transforms raw data into a more understandable format. It plays a crucial role in data analysis and machine learning processes.

e.g : Facebook dataset has two feature of vectors types:

- Categorical features: such as name, Intro,Profile Picture, Living Place,Check-In .
- Numerical features: such as Likes,Mutual friends , Groups,Posts,Tags .

We processed the dataset by applying classification algorithms and considering the types of numerical features. Additionally, we converted categorical features into numerical representations for analysis.

To streamline our analysis, we focused on testing the most significant attributes from the dataset. Non-significant attributes such as Link and Name-Id (in the case of Facebook dataset) were excluded from our model. Instead, we replaced Name-Id with a numerical identifier for each example. This preprocessing step was crucial for applying various machine learning algorithms to the dataset.

To ensure the accuracy of the dataset, a filtering process is applied. The classification algorithm effectively categorizes the dataset when it is devoid of incorrect or null values.

As a part of the data preprocessing phase, we performed data normalization. This step is crucial to maintain the information integrity by scaling down the large numerical values to a common range of [0, 1] without distorting the relative differences between values. Certain

algorithms necessitate this phase for accurate modeling of the data.

Normalization involves scaling selected attributes from their original values to a range of 0 to 1. This process is performed when the dataset features have varying measurement units, when the data distribution is unclear or random, or when the distribution is non-Gaussian. Normalization is applied to ensure uniformity and comparability across different attributes in the dataset.

4.2.3 Features selection

We use feature selection when there are redundant features in the dataset or the features are very insignificant for the results.[22]

Correlation matrix

The **correlation matrix** was employed to identify the attributes and classes with the strongest correlation. The correlation matrix provides a comprehensive view of the correlation coefficients between various variables. Each cell in the matrix represents the correlation between two variables, serving as a summary of data, input for advanced analysis, and a diagnostic tool for further investigations.

This is done to understand the relationship between two variables and the strength of association betweenthem. We calculated the correlation matrix and concluded absence of high multicollinearity between the variables. [26]

4.2.4 Cleaning and scaling

Missing values are a common issue in datasets, arising from various real-world issues. They can be addressed through either deletion or imputation methods. Missing values have a negative

impact on the amount of data available for analysis, reducing the statistical power of a study and potentially compromising the validity of its findings.

Due to a missing value in the "Link" attribute, one row is excluded from the dataset table. For numerical attributes, the missing values can be replaced with either 0 or the mean value of the other values within the same row.

Rescaling the data helped the machine learning models to perform better(using the library **StandarScaler** .

We proceeded with the dataset by considering multiple classification algorithms and giving careful consideration to numerical feature types. Additionally, we transformed the numerical aspects of other categorical features. Given the dataset's numerous attributes, we focused on testing the most significant ones and excluded insignificant attributes like Name-Id from our model. The application of various machine learning algorithms to the dataset is of utmost importance.

Filtration is applied to ensure accurate classification of the dataset. If the dataset is free from incorrect or null values, the classification algorithm will correctly classify the dataset.

As part of the data preprocessing phase, we performed data normalization to preserve information and ensure that dispersed numerical values are transformed to a common scale of [0, 1]. This step is crucial for maintaining the relative differences in value ranges without distortion. Certain algorithms require this normalization phase to effectively model the data. When the data distribution is random, normalization proves to be a beneficial strategy, as it improves coefficients by rescaling selected attributes to a scale of 0 to 1 after training. Once data pre-processing is done, we can safely move into the algorithms.

4.2.5 Training fake profile detection models

Cross-Validation :

Cross-validation (CV) is a statistical technique used to assess the effectiveness of machine learning models. It involves dividing the data into two sections: one for training the model and the other for model validation. By splitting the data and evaluating the model's performance on multiple subsets, CV allows for a more comprehensive assessment and comparison of different learning algorithms.

Data Classification :

- We applied the two classification type (**supervised and unsupervised learning**) to compare their results .
- Machine learning programs utilize diverse algorithms and pre-categorized training datasets to classify upcoming accounts as either fake or real.
- We divide the data into two sets: a Training set and a Testing set.
- The training set is shown to our model, and the model learns from the data in it.
- The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.
- We have used a test-train split of **20% -80% and 40%-60%** to compare the performance of machine learning algorithms for the predictive modeling fake detection problem .
  - Train Dataset: Used to fit the machine learning model.
  - Test Dataset: Used to evaluate the fit machine learning model.

- The aim is to evaluate the performance of the machine learning model on unseen data, which refers to data that was not used during the model's training process.

- In situations where there is insufficient data, the k-fold cross-validation procedure serves as an effective alternative model evaluation method.

- This quantitative case study explores the empirical correlation between features like likes, profile picture, relationship status, mutual friends, and employment (in the context of the Facebook dataset) with the likelihood of being a fake user.

Parameters tuning

For each model, specific parameters were carefully chosen and assigned a range of potential values. These parameters are crucial for effectively detecting illegitimate accounts and determining the learning rate. They will be integrated into bio-inspired algorithms as part of the implementation process.[20]

4.2.6 Testing fake profile detection models

Machine learning models

This section provides a detailed introduction to our suggested model, which is composed of three primary stages: data prepossessing, data reduction, features selection, and data classification. Our endeavor commenced with the processing of the dataset, followed by the incorporation of various reduction techniques during the subsequent phase. Within the reduction stage, the data underwent filtration and reduction utilizing specific mechanisms, preparing it for the ensuing classification phase. During this classification phase, the refined data underwent classification algorithms, ultimately revealing the conclusive outcomes.

We have used in our comparative study the next supervised models :Linear regression ,Decision Tree Model, Support Vector Machine , Random Forest Model,KNearest Neighbors. And we've choose the k-means model as an unsupervised technique

Python was employed to construct the test scenario for assessing diverse machine learning methods in deploying the earlier elucidated techniques.

With the aim of investigating accessible datasets of counterfeit accounts using diverse statistical methods and applying machine learning algorithms, this study takes the form of a **quantitative case study**. The study commenced by initially modeling the dataset without feature selection. All experiments incorporated techniques such as k-fold cross-validation to mitigate over-fitting, along with parameter tuning to identify optimal parameters for the employed model (in the case of the KNN model).

Implementation of Algorithm

- Step 1: Load and read the datasets
- Step 2: Clean the data by filling the missing values
- Step 3: Divide the all dataset in two parts: Test dataset and Train dataset
- Step 4: Apply different machine learning techniques
- Step 5: Generate the confusion matrix of each technique
- Step 6: Compare the values of evaluation parameters of each technique and analyze the results

Parameter Tunning

1. **Supervised algorithms :**

   Linear regression:test size=(0.2.0.4) , train size=(0.8,0.6)

   Decision Tree:Random state=42 , test size=(0.2.0.4) , train size=(0.8,0.6)

   Support Vector Machine:karnel=linear , test size=(0.2.0.4) , train size=(0.8,0.6)

   Random Forest Machine :n estimators=100, random state=42,test size=(0.2.0.4) ,train size=(0.8,0.6)

   K-nearest Neighbors Algorithm: test size=(0.2.0.4) , train size=(0.8,0.6)

   The number of neighbors to check in a KNN model is implemented for KNN hyperparameter tuning as (k=2/k=3/k=4/k=5/k=6/K=7/K=8/K=9).

2. **Unsupervised algorithms :**

   We have two versions of the K means Model for unsupervised algorithms: Canonical K means: Number of clusters=3 ,test size=(0.2) , train size=(0.8) Manual K means: max iterations = 100 , number of clusters =3.test size=(0.2) , train size=(0.8)

3. **Bio inspired algorithms :**

   GWO:

   The number of wolves or solutions in the population. Default is 10.

   The maximum number of iterations or generations. Default is 50.

   alpha: The alpha parameter of the GWO algorithm, controlling the influence of the alpha wolf. Default is 0.5.

   a: The parameter determining the number of top solutions (alpha, beta, delta) to update in each iteration. Default is 2

MFO: The number of moths or solutions in the population. Default is 15.

The maximum number of iterations or generations. Default is 100.

The threshold value for moth brightness. Default is 0.1.

The parameter controlling the influence of the flame attraction. Default is 1.0.

The parameter controlling the decay rate of the flame attraction. Default is 1.0.

The lower bound of the decision variables. If not specified, it is initialized with zeros.

The upper bound of the decision variables. If not specified, it is initialized with ones.

GOA:

The number of grasshoppers in the population. Default is 10.

The maximum number of iterations. Default is 100.

The attraction coefficient. Default is 1.0.

The repulsion coefficient. Default is 1.0.

The lower bounds for the grasshopper positions. If not specified, it is initialized to zeros.

The upper bounds for the grasshopper positions. If not specified, it is initialized to ones.

WOA:

The number of search agents (whales) in the algorithm.Default 10.

The maximum number of iterations.Default 50.

The lower bounds for the grasshopper positions. If not specified, it is initialized to zeros.

The upper bounds for the grasshopper positions. If not specified, it is initialized to zero.

4.3 Fake profile detection approach

4.3.1 Motivation

Animal-inspired optimization algorithms like the Grasshopper Optimization Algorithm is promising approaches to solving large-scale optimization problems. They are useful because they do not require prior knowledge of the problem and can be applied to various fields. By mimicking the social behavior of animals, these algorithms use random and guided movements to find the best possible solutions.

In this part We've chosen five bio-inspired algorithms: Grey Wolf Optimizer(GWO), Whale Optimization Algorithm (WOA), Moth Flame Optimizer(MFO), and Grasshopper Optimization Algorithm (GOA), to understand their performance based on different performance metrics , and compares with algorithms of machine learning .

4.3.2 Chosen performance evaluation metrics

Various performance metrics including F1 score, confusion matrix and recall can do this Used to evaluate the identification of fake accounts. As performance indicators for our research: We used ACC (Accuracy), F-Score, Recall and Precision. Counterfeit Model Detection Use a confusion matrix to visualize accounts.

- TP = True Positives, when our model correctly classifies the data point to the class it belongs to.
- FP = False Positives, when the model falsely classifies the data point.
- TN = These are the cases where the predicted "No" actually belonged to class "No".

- FN = These are the cases where the predicted “No” actually belonged to class “Yes”.[28]

- **Precision** is used to calculate the model's ability to classify values correctly. It is given by dividing the number of correctly classified profiles by the total number of classified data points for that class label.

- **Recall** is used to calculate the ability of the model to predict positive values. But, ”How often does the model predict the correct positive values?”. This is calculated by the ratio of true positives and the total number of actual positive values.

- **F1-score** should be used when both precision and recall are important for the use case. F1 score is the harmonic mean of precision and recall. It lies between [0,1].[28]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = Sensitivity = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.4}$$

4.4 Our system :

Our system follows the following steps:

Dataset: We start with a dataset comprising pertinent information tailored for our task—specifically, profiles for detection.

Data Preprocessing: Data preprocessing is an important step in preparing raw data Analysis and model training. Here are the steps in summary:

- Data Cleaning: Commencing, we cleanse the data by eliminating extraneous or disruptive details. This encompasses rectifying missing entries, resolving disparities, and addressing outliers. This cleanliness assurance curtails the likelihood of introducing biases or inaccuracies in ensuing procedures.

- Data Management: Data management encompasses arranging and structuring data for streamlined analysis. This incorporates standardizing formats, ensuring uniformity across diverse sources, and managing data disparities.

- Data Transformation: Data transformation entails altering data to render it suitable for analysis and model training. This could encompass actions like feature scaling, normalization, or log transformations, aligning the data with assumptions necessary for algorithms employed in subsequent stages.

- Data Integration: In some cases,data might originate from various datasets or disparate origins. Data integration entails merging and amalgamating these datasets, guaranteeing alignment through pertinent attributes or keys. This phase enables us to utilize an all-encompassing dataset for analysis and modeling.

- Data Reduction: Data reduction methods strive to diminish the dataset's dimensionality by removing superfluous or insignificant attributes. This streamlines the dataset, enhances

computational efficiency, and mitigates the danger of overfitting in subsequent phases.

- Feature Selection: Feature selection encompasses choosing a pertinent subset of features that significantly contribute to prediction or analysis objectives. This approach aids in simplifying complexity, boosting interpretability, and enhancing model performance.

Data Split: After preprocessing, the dataset is divided into two subsets: the training set and the dataset a test set. The training set is used to train our contour detection model while using the test set Used to evaluate model performance.

Training with BIA/ML: The training set is employed to train our profile detection model using either Bio-Inspired Algorithms (BIA) or Machine Learning (ML) techniques. BIA involves creating algorithms inspired by biological systems, while ML encompasses a spectrum of algorithms capable of learning patterns and making predictions from data.

Trained Model: Once the training process is completed, we obtain a trained model that has learned from the training data. This model is capable of detecting profiles based on the learned patterns and characteristics.

Test with BIA/ML: The test set gauges our trained model's efficacy. By applying the model to the test data, we compare its predictions against actual labels to measure accuracy and efficiency in profile detection.

Profile Detection: The trained model is deployed on novel, unseen data or real-world contexts for profile detection. It scrutinizes input profiles, leveraging patterns and attributes learned during training, to discern their authenticity and potential fakeness.

Validate Performance: The profile detection system's performance is validated through a comparison of the model's predictions with established profiles. This process encompasses evaluating

metrics like accuracy, precision, recall, and F1 score to gauge the system's efficacy in identifying both authentic and counterfeit profiles.

Through this approach, we adeptly identify profiles by synergizing data preprocessing, BIA/ML training, and assessing the profile detection system's efficacy.

4.5 Transition from natural to artificial:

4.5.1 The Grey Wolf Optimizer (GWO)

In the context of the Grey Wolf Optimization algorithm, the mentioned points can be explained as follows:

- The Grey Wolf joins its pack after they have found the prey:

  The search for the optimal solution occurs in the gray wolf optimization algorithm ,Inspired by the hunting behavior of gray wolves.Wolves pack together and work together Find and catch prey. Also in the optimization algorithm gray wolf (candidate solutions) work together and share information to improve their search Optimal solution. Once the wolf finds a promising solution, it shares it Information related to the rest of the package

- Each candidate solution is classified into the most appropriate class (Real or Fake):

  For some optimization problems, it is necessary to classify candidate solutions into Different categories are divided according to certain criteria. These classes can represent different types or solution category. For example, in our case, in the binary classification problem, These classes can be ”true” and ”fake”.

- Suppose there are two points of prey in the search space:

  In the context of the optimization algorithm, the search space embodies all potential problem solutions. The term ”prey” alludes to the sought-after, ideal solution. This notion implies that within the search space, two distinct optimal solutions exist, both of which the grey wolves are striving to discover.

- Two classes (Real or Fake):

  The optimization problem considered is to divide the solution into two classes Category: ”Real” or ”Fake”. This classification may be based on certain restrictions, The goal or feature of the problem.

- Environment: Online social network (Facebook, Twitter, Instagram):

  The term ”environment” refers to the context or domain in which the optimization is performed Question applied. In this example, the environment is an online social network, This could be a platform such as Facebook, Twitter or Instagram.

- Grey wolf: Online social network user:

  Each grey wolf corresponds to an individual user within the online social network, and each grey wolf's behavior is modeled after the hunting behavior of grey wolves in nature.

- Group of grey wolves: Online social network users:

  The pack of grey wolves in the optimization algorithm represents a group of online social network users who collaborate to find the best solution to the optimization problem by sharing knowledge, tactics, and solutions.

- Best individual of each pack of Grey Wolves: The best solution is alpha, which stores the position of the alpha wolf:

  The best solution represents the most promising candidate within the pack and stores the position or configuration associated with the optimal solution found by that pack in the optimization algorithm. This best solution is often referred to as the "alpha" wolf.

- The distance between the prey and the spotted Grey Wolf: D = |C * pos - wolves[i]| represents the distance between the current position of a wolf and the target position being updated:

  The Grey Wolf Optimization algorithm uses the distance between a wolf and the prey (optimal solution) to direct the search process. The equation D = |C * pos

  - wolves[i]| calculates the distance between the current position of a wolf (wolves[i]) and the target position being updated. This distance calculation is used to update the positions of the wolves during the optimization process, allow them to move closer to the target solution.

4.5.2 The Whale Optimization Algorithm (woa)

In the context of the Whale Optimization Algorithm (WOA), the corresponding points can be explained as follows:

- Whales operating individually in the search space: Each whale represents a candidate solution and conducts an independent search within the search space to optimize the objective function in the WOA.

- Suppose there are two points of prey in the search space: Two potential optimal solutions (prey) are present in the search space of the optimization problem, which the whales are trying to find.

- Two classes (Real or Fake): This classification helps evaluate and categorize the solutions based on their fitness or suitability within the problem context and allows the candidate solutions in the WOA to be divided into two classes, such as ”Real” or ”Fake”.

- Environment: Online social network (Facebook, Twitter, Instagram): The environment represents an online social network context, such as Facebook, Twitter, or Instagram.

- Whale: Online social network user: In this case, each whale represents an individual user of an online social network, and each whale's behavior is modeled after the traits or actions of whales in the wild.

- Group of Whales: Online social network users: A collective of users of online social networks, the whales in the WOA operate independently within the environment of these networks and modify their tactical approaches to collectively optimize the objective function.

- The ”best position” variable will contain the response that obtained the highest fitness value discovered during the optimization process:

  The solution represents the optimal or best solution discovered by the algorithm for the given objective function and search space constraints, and is stored in the variable called the ”best position” that is maintained by each pack of whales.

- The distance between the whale and its prey:

The parameter C is a coefficient that controls the exploration and exploitation balance in the algorithm, and it is used to calculate the distance between a whale and its prey (represented by the best position) in the WOA. The distance calculation is used to direct the movement of whales towards the target solution, and by updating the positions based on the distance calculation, the whales converge towards the best solution.

4.5.3 The Moth Flame Optimizer (MFO)

In the context of the Moth Flame Optimization (MFO) algorithm, the corresponding points can be explained as follows:

- Operating individually in the search space:

  Each moth in the MFO algorithm represents a candidate solution, and each moth operates independently within the search space to optimize the objective function.

- The most suitable class (Real or Fake) is assigned to each candidate solution:

  The MFO algorithm can classify the candidate solutions into two classes, such as ”Real” or ”Fake”, which helps evaluate and categorize the solutions based on their fitness or suitability within the problem context.

- Suppose there are two points of prey in the search space: The moths are looking for two potential optimal solutions (prey) in the search space of the optimization problem.

- Two classes (Real or Fake): Based on their traits or fitness values, the candidate solutions can be divided into two classes, typically ”Real” or ”Fake.”

- Environment: Online social network (Facebook, Twitter, Instagram): The environment represents an online social network context, such as Facebook, Twitter, or Instagram.

- Moth: Online social network user: Each moth represents an individual user of an online social networking site, and their actions are modeled after the traits or conduct of moths in the natural world.

- Group of Moths: Online social network users: A collective of users of online social networks is represented by the group of moths in the MFO algorithm.

- Best individual of each pack of Moths: The moth is the best solution or position found by the MFO algorithm after the specified number of iterations: The solution represents the optimal or best solution discovered by the algorithm after a certain number of iterations and is stored in a variable called the "best solution" or "best position" that is maintained by each pack of moths.

- The distance between the moth and the light source: The Euclidean distance formula is used in the MFO algorithm to calculate the distance between a moth's position (moths[i]) and the position of the light source (flame), which helps determine the moth's proximity to the light source and directs the movement of moths towards the target solution.

4.5.4 Grasshopper Optimization Algorithm (GOA)

In the context of the Grasshopper Optimization Algorithm (GOA), the corresponding points can be explained as follows:

- Grasshoppers iteratively explore the search space, leveraging attraction and repulsion mechanisms to guide their movements towards better solutions: Grasshoppers iteratively

explore the search space by balancing attraction towards better solutions and repulsion to avoid overcrowding. This mechanism helps guide their movements towards optimal or near-optimal solutions. The Grasshopper Optimization Algorithm simulates the behavior of grasshoppers in nature to optimize a given objective function.

- Each candidate solution is classified into the most appropriate class (Real or Fake): Similar to earlier explanations, candidate solutions in the GOA can be categorized into various classes, such as "Real" or "Fake," based on specific criteria or characteristics. This classification aids in assessing and choosing the best solutions during the optimization process.

- Suppose there are two points of prey in the search space: The grasshoppers try to find these points during the optimization process in the GOA, where there can be two points of prey or optimal solutions within the search space.

- Two classes (Real or Fake): In the GOA, candidate solutions can also be divided into two groups, typically "Real" or "Fake," based on how well they fit the parameters of the problem.

- Environment: Online social network (Facebook, Twitter, Instagram):

  In this instance, the environment is an online social network environment, like Facebook, Twitter, or Instagram.

- Grasshopper: Online social network user: Each grasshopper in the GOA represents an individual user of an online social network, and each grasshopper's behavior is modeled after the traits or actions of natural grasshoppers.

- Group of Grasshoppers: Online social network users: Users of online social networks are represented by the grasshoppers in the GOA.

- The best grasshopper in each group of grasshoppers represents the position at which the Grasshopper Optimization algorithm determined to be the optimal solution to the given optimization problem:

  The grasshopper that achieves the highest fitness value or offers the best solution among all the grasshoppers in the population is referred to as the best individual in the GOA, and this best grasshopper represents the best solution determined by the GOA algorithm for the given optimization problem.

- The distance between a grasshopper and its desired location, which influences its choice of movement and behavior:

  The Euclidean distance formula is used to calculate the distance between the current grasshopper's position (p) and another grasshopper's position (q) in the search space, which influences the grasshoppers' choice of movement and behavior by indicating how close the two positions are to each other.

4.6 Bio-Inspired Algorithms' Purpose and Machine Learning Algorithms Comparison:

The choice of using a regression logistique model (LR) in an optimization algorithm depends on the context of our problem and our specific goals. In this case, we have: Optimisation de Paramètres. The models of regression logistique are frequently used to optimize parameters or coefficients.

bio-inspired algorithms like GWO, MFO, GOA, and WOA have mathematical function optimization,named fitness function calculates this function by combining a machine learning model like LR, which we have chosen in this case, with the bio-inspired algorithms. This function takes the parameters from the bio-inspired algorithms and feeds them into the chosen model.

After running the bio-inspired algorithms, we obtain the best solutions. For example, if the Facebook dataset contains 1025 rows, the best solutions may comprise 520 rows, for instance. this is is the best solution

Then, we summarize the dataset by selecting only the optimal solutions and apply the chosen model "LR" with the parameters obtained from the best solutions.

4.7 Experimental software environment

4.7.1 Testing software environment

This study makes use of a variety of tools. They're all open source and free.

- Python 3.5
- NumPy 1.11.3
- Matplotlib 1.5.3
- Pandas 0.19.1
- SciPy and Scikit-learn 0.18.1
- Mealpy 2.4.0
- Jupyter Notebook

Python [2] is a high level general programming language and is very widely used in all types of disciplines such as general programming, web development, software development, data analysis, machine learning etc. Python is used for this project because it is very flexible and easy to use and also documentation and community support is very large.

NumPy [3] is very powerful package which enables us for scientific computing. It comes with sophisticated functions and is able to perform N-dimensional array, algebra, Fourier transform etc. NumPy is used very where in data analysis, image processing and also different other libraries are built above NumPy and NumPy acts as a base stack for those libraries

Pandas [4] is open source BSD licensed software specially written for python programming language. It provides complete set of data analysis tools for python and is best competitor for R programming language. Operations like reading data-frame, reading csv and excel files, slicing, indexing, merging, handling missing data etc., can be easily performed with Pandas. Most important feature of Pandas is, it can perform time series analysis

SciPy[5]is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data. With SciPy, an interactive Python session becomes a data-processing and system-prototyping environment rivaling systems, such as MATLAB, IDL, Octave, R-Lab, and SciLab.

For this study, scikit-learn is used because it is based on python and can interoperate to

[2]https://www.python.org/
[3]https://numpy.org
[4]https://pandas.pydata.org
[5]https://docs.scipy.org/doc/scipy/tutorial/general.html

NumPy library. It is also very easy to use.

Scikit-Learn (SKLearn) [6] is an environment that is integrated with Python programming language. The library offers a wide range of supervised algorithms . The library offers high-level implementation to train with the 'Fit' methods and 'predict' from an Classifier and also offers to perform the cross validation, feature selection and parameter tuning.

Mealpy [7]is a Python library for the most of cutting-edge population meta-heuristic algorithms - a field which provides an efficient way to find the global optimal point of mathematical optimization problems.

Jupyter Notebook[8]is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

4.8 Conclusion

The results indicate that the clustering quality is improved compared to the standard random selection of initial centroids . We also experimentally compare our method with the other evolutionary proposed (GA) for initial centroid selection and the experimental results show that our method performs better in most cases.

[6]https://scikit-learn.org
[7]https://mealpy.readthedocs.io
[8]https://jupyter.org/

Chapter 5

Results and discussion

5.1 Introduction

This section presents results examining our dataset and a comparative analysis of the algorithms. After all preprocessing, descriptive and exploratory analysis, the dataset was deployed on different machine learning and bio inspired algorithms .

The experimental results are shown in the figure The tables and explanations below present the best Performers according to various performance indicators.

The following tables shows the results: we'll present the results organized by datasets as follow :

1. Without normalization

2. With normalization

3. With features selection

Each of these subsections is presented by the next taxonomy:

- Supervised techniques .

- Unsupervised techniques .

- Bio inspired Algorithms .

5.2 Machine Learning algorithms results

5.2.1 Facebook dataset

1. **Supervised Techniques :**

The table 5.1 below summarizes the calculated metric for supervised training models on facebook dataset without normalization :

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.9435	0.8798	0.9665	0.9098	0.8845	0.7843	0.8023	0.7976
DT	0.9659	0.9254	0.9546	0.9476	**0.9865**	**0.9657**	**0.9743**	**0.9687**
SVM	0.9859	0.9665	0.9989	0.9779	0.9876	0.9665	0.9557	0.9698
RF	**0.9912**	**0.9742**	**0.9887**	**0.9887**	0.9154	0.8243	0.8789	0.8476
KNN(K=2)	0.9061	0.7645	0.9165	0.8165	0.8443	0.7456	0.8487	0.7754
KNN(K=3)	0.9045	0.8265	0.8778	0.8467	0.8867	0.7978	0.7198	0.7467
KNN(K=4)	0.8888	0.7832	0.8290	0.8060	0.8587	0.7375	0.8087	0.7587
KNN(K=5)	**0.9247**	**0.8145**	**0.8698**	**0.8334**	0.8765	0.7789	0.7465	0.7589
KNN(K=6)	0.8998	0.8267	0.7976	0.8035	0.8487	0.6987	0.7287	0.7187
KNN(K=7)	0.8725	0.7689	0.6854	0.7165	0.8990	0.8376	0.7476	0.7865
KNN(K=8)	0.8888	0.8609	0.7543	0.7954	0.8598	0.7354	0.7287	0.7243
KNN(K=9)	0.8943	0.8287	0.7468	0.7743	0.8565	0.7643	0.6865	0.7145

Table 5.1: Evaluation of supervised algorithms for Facebook dataset

Following RF and DT, Logistic Regression (LR), K-Nearest Neighbors (KNN), and Sup-

port Vector Machine (SVM) is produced reasonably good results. It's important to note that . the results of the experiment show that both Random Forest (RF) and Decision Tree (DT) algorithms performed exceptionally well in the two train-test splits. RF showed the best performance in the 80%/20% training and testing split, and DT performed the best in the 60%/40% training and testing split.

Additionally, the K-Nearest Neighbors (KNN) algorithm was evaluated, and it was found that it performed best at k=5. This finding suggests that choosing an appropriate value for the hyperparameter k can significantly impact the performance of the KNN algorithm, so it is important to consider hyperparameter tuning for KNN to optimize its performance in the given task.

The Figure 5.1 shows the representation of accuracy of different supervised models.

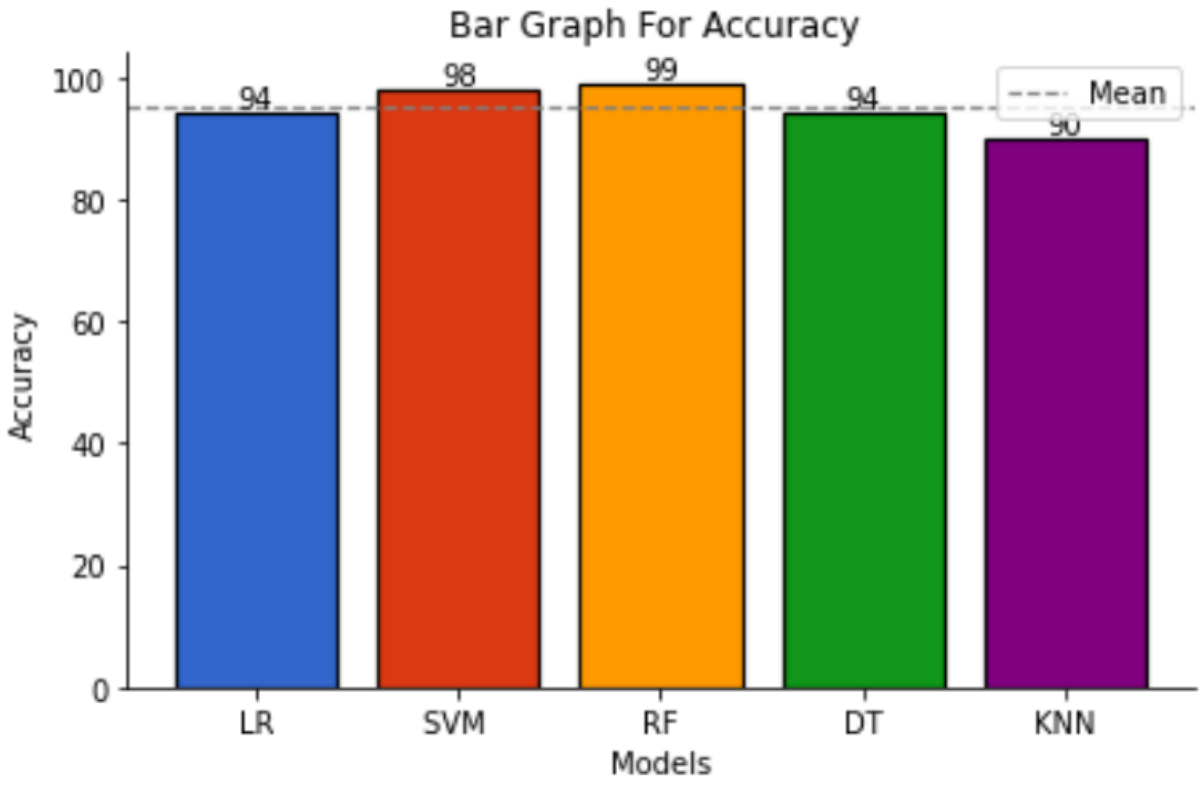


Figure 5.1: *Bar graph of Accuracy for supervised algorithms*

The accuracy ratings of various supervised learning algorithms are shown in the graphe. It is clear that Random Forest (RF) compared to other algorithms and Support Vector Machine (SVM) algorithms have the highest accuracy.

The graph serves as visual evidence, highlighting the higher accuracy performance of RF and SVM. This knowledge helps in decision making as it means that RF and SVM algorithms are the best choice when accuracy is important for a predictive model.

2. **Unsupervised Techniques :**

The table 5.2 shows the evaluation metric for both random and manual k-means model on Facebook dataset. We can mark that both random and manual k-means gave worst results.

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	**0.5198**	**0.5175**	**0.5287**	**0.4567**
Manuel K-means	0.4867	0.4944	0.4965	0.4268

Table 5.2: Evaluation of unsupervised algorithm for Facebook dataset

Table 5.2 shows the scoring scales for random and manual K-Means models in the Facebook dataset. It's clear that both models received poor and low ratings, indicating sub-par performance. But it should be noted that the value of the conventional k-means model is slightly higher than that of the manual k-means model.

Features selection

- Correlation heatMap of Facebook dataset

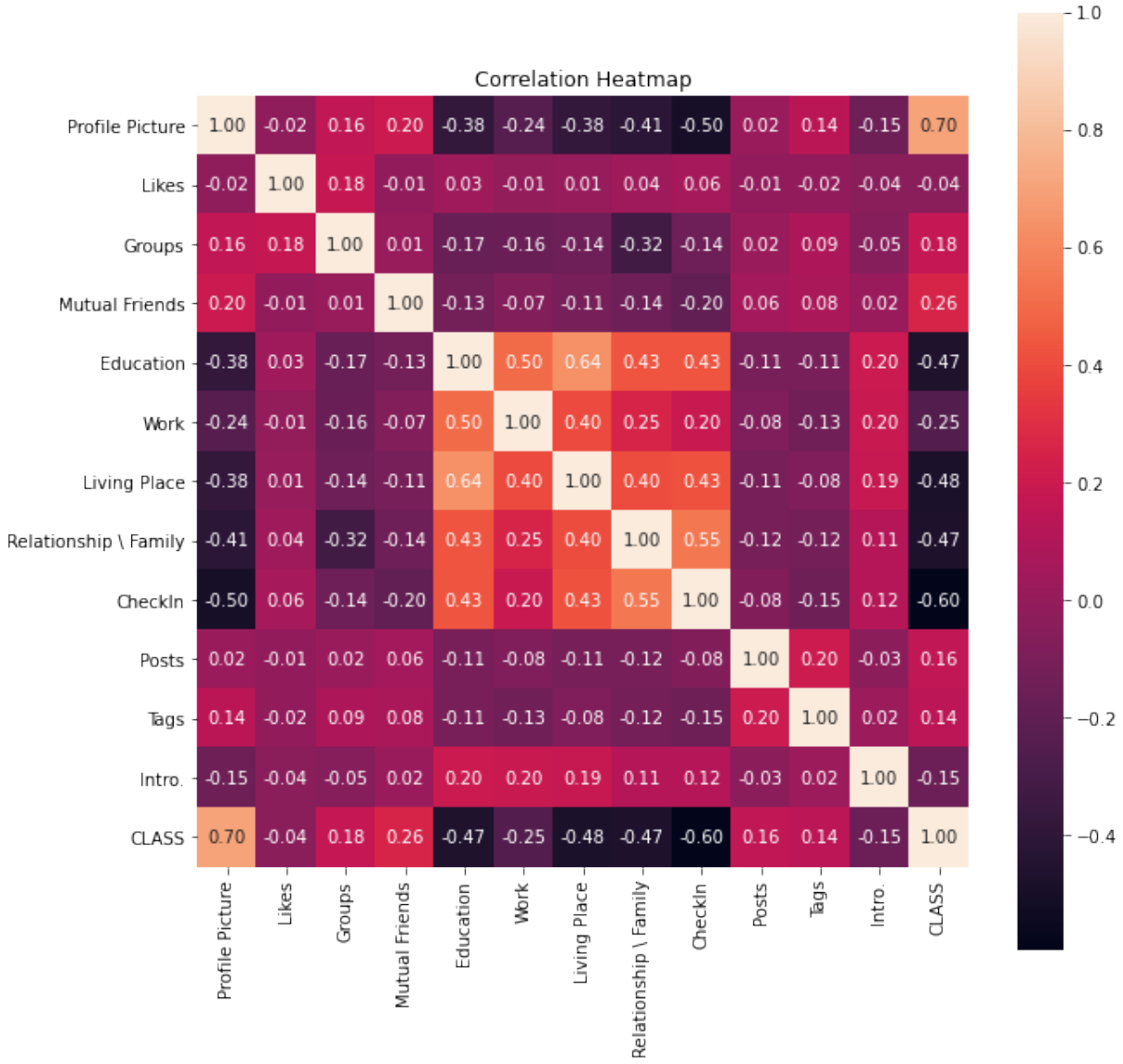


Figure 5.2: *Correlation heatMap of Facebook dataset*

In the figure 5.2, the correlation matrix between the dataset features is showed. We can see that there are some features that have a highest correlation with the target class (Profile Picture, CheckIn, Living Place, Relation ship and family) and that will help us in features selection for better performance results .

1. **Supervised Techniques :**

The performance results for supervised algorithms after features selection on Facebook dataset can be found in the table 5.3 .

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.9332	0.8346	0.9349	0.8794	O,9568	0.8967	0.9498	0.8189
DT	0.9543	0.9099	0.9105	0.9049	0.9578	0.8865	0.9476	0.9176
SVM	0.9686	0.9273	0.9693	0.9476	0.9475	0.9165	0.8887	0.8943
RF	0.9545	0.9306	0.9174	0.9254	0.9345	0.8676	0.9165	0.8867
KNN(K=2)	0.9336	0.8468	0.9137	0.8736	0.9557	0.8954	0.9389	0.9165
KNN(K=3)	0.9574	0.9264	0.9058	0.9164	0.8434	0.7876	0.5377	0.5154
KNN(K=4)	**0.9739**	**0.9375**	**0.9647**	**0.9483**	0.9575	0.9044	0.9365	0.9132
KNN(K=5)	0.9496	0.8776	0.9568	0.9092	0.9468	0.8935	0.9378	0.9145
KNN(K=6)	0.9573	0.8865	0.9543	0.9185	0.9475	0.8968	0.9487	0.9078
KNN(K=7)	0.9647	0.9046	0.9186	0.9093	**0.9654**	**0.9074**	**0.9666**	**0.9376**
KNN(K=8)	0.9453	0.8897	0.8936	0.8971	0.9575	0.8878	0.9389	0.9054
KNN(K=9)	0.9754	0.9497	0.9235	0.9384	0.9196	0.8765	0.7887	0.8298

Table 5.3: Evaluation of supervised algorithms for Facebook dataset (after features selection)

The experimental results show that the classification effect is significantly improved, The accuracy of the dataset is tested using the proposed feature selection strategy. These results highlight the effectiveness of the feature selection strategy in selecting relevant features for accurate predictions and ultimately enhancing the performance of classification models. This improvement is particularly noticeable when using specific algorithms, such

as K-Nearest Neighbors (KNN) in both train-test splits. However, it should be noted that the accuracy values for random forest (RF)and decision tree (DT).

2. **Unsupervised Techniques :**

   The results of unsupervised algorithms after features selection on facebook dataset are shown in table 5.4 The k means model gave bad performance results.

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	**0.8534**	**0.7535**	**0.8954**	**0.7932**
Manuel K-means	0.5076	0.6165	0.5734	0.4846

Table 5.4: Evaluation of unsupervised algorithm for Facebook dataset (after features selection)

After feature selection on the Facebook dataset, the results of the unsupervised algorithm show a significant increase in accuracy. More precisely, both Canonical k-means and manual k-means algorithms show commendable performance. But it should be emphasized that the conventional k-means model is used outperforms manual K-Means models and achieves the highest accuracy in comparison.

Data normalization

Because our Facebook dataset has varying scales we have scaled the dataset in range between 0 and 1 to give equal weights/importance to each variable.

1. **Supervised Thechniques :**

The table 5.5 presents the performance results on normalized data with the supervised algorithms.

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.9665	0.9198	0.9578	0.9398	0.9534	0.8845	0.9534	0.9109
DT	**0.9834**	**0.9535**	**0.9888**	**0.9676**	**0.9857**	**0.9667**	**0.9756**	**0.9688**
SVM	0.9732	0.9045	0.9489	0.9254	0.9689	0.9190	0.9478	0.9265
RF	**0.9813**	**0.9535**	**0.9676**	**0.9735**	**0.9808**	**0.9787**	**0.9698**	**0.9743**
KNN(K=2)	0.9321	0.8754	0.9454	0.9067	0.9265	0.8443	0.9276	0.8756
KNN(K=3)	0.9534	0.9235	0.9389	0.9279	0.9443	0.8921	0.8889	0.8878
KNN(K=4)	**0.9765**	**0.9379**	**0.9676**	**0.9476**	0.9556	0.8945	0.9409	0.9198
KNN(K=5)	0.9445	0.9223	0.9089	0.9165	0.9578	0.8867	0.9176	0.8976
KNN(K=6)	0.9564	0.8809	0.9176	0.8954	0.9565	0.9065	0.9332	0.9167
KNN(K=7)	0.9698	0.9343	0.9553	0.9467	0.9443	0.8832	0.8945	0.8956
KNN(K=8)	0.9443	0.8998	0.9032	0.9089	0.9434	0.8945	0.9167	0.9043
KNN(K=9)	0.9245	0.8523	0.8768	0.8665	**0.9645**	**0.9167**	**0.9389**	**0.9232**

Table 5.5: Evaluation of supervised algorithms for normalized Facebook dataset)

Decision Trees (DT) and Random Forests (RF) are invariant to data scaling as they base their decisions on feature values rather than their magnitudes. However, proper feature scaling (normalization) is essential for the K-Nearest Neighbors (KNN) classifier. KNN relies on distance calculations, and if features have varying scales, larger values can dom-

inate the distance calculation and introduce bias into the classification. By standardizing the feature values through normalization, KNN's performance improves significantly. This standardization ensures that each feature carries equal importance and eliminates the influence of different value ranges. Consequently, while DT and RF do not necessitate scaling, standardizing feature values positively affects KNN's classification accuracy.

2. **Unsupervised Techniques :**

The table 5.6 presents the performance results on normalized data with the unsupervised algorithms.

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	0.1467	0.2498	0.1176	0.1387
Manuel K-means	**0.8276**	**0.4256**	**0.4887**	**0.4556**

Table 5.6: Evaluation of unsupervised algorithm for normalized Facebook dataset)

From the table 5.6, we can remark the K-means model does not perform well even if we normalize the data. After analyzing the data in Table 5.6, it is clear that the K-means model's performance is unsatisfactory, even with data normalization techniques. Surprisingly, the accuracy value even decreases further. Interestingly, the manual K-means algorithm shows a slight advantage over the canonical K-means model.

Comparative results on Facebook dataset:

The article [3] and this work share the common objective of detecting fake content using machine learning algorithms. However, there are some key differences in the approaches taken in each work. The two both works focus on the same dataset for detecting fake Facebook profiles. - The results indicate that supervised algorithms outperform unsupervised algorithms in terms of accuracy rates in both studies. - The k-Means and k-Medoids algorithms yield the best results when applied to numerical attributes exclusively. Regarding the non-common results, despite differences in approaches, our work outperforms in Decision Tree, Support Vector Machine, and K-Nearest Neighbors (with k=5).

Table 5.7 represent a Comparative our work in "Facebook dataset" with another article

	[3]	Our Approche
	Accuracy	
DT (ID3)	97	98
SVM	95	97
KNN	91	97

Table 5.7: Comparative our work in "Facebook dataset" with another article)

In Table 5.6, a comparison is made between our work on the "Facebook dataset" and the findings presented in the article "A Machine Learning Model for Detecting Fake." The results unambiguously demonstrate that our work attains higher accuracy than the mentioned article across all three algorithms: Decision Tree (DT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).

5.2.2 Twitter dataset

1. **Supervised Techniques**

The table 5.8 shows the metric for supervised training models on twitter dataset :

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.8912	0.9067	0.8923	0.8934	0.8723	0.8667	0.8745	0.8723
DT	0.9046	0.8987	0.9187	0.8954	0.8915	0.8787	0.8765	0.8744
SVM	0.8132	0.8034	0.9198	0.7965	0.8054	0.7987	0.7980	0.7956
RF	**0.9300**	**0.9223**	**0.9376**	**0.9243**	**0.9246**	**0.9209**	**0.9087**	**0.9276**
KNN(K=2)	0.7925	0.8143	0.7909	0.7934	0.7897	0.7976	0.7854	0.7787
KNN(K=3)	0.8132	0.8043	0.8076	0.8067	0.8343	0.8254	0.8468	0.8598
KNN(K=4)	0.8365	0.8313	0.8254	0.8065	0.7924	0.7935	0.7898	0.7876
KNN(K=5)	0.8478	0.8532	0.8434	0.8443	0.8198	0.8232	0.8165	0.8154
KNN(K=6)	0.7656	0.7576	0.7665	0.7523	0.7955	0.8068	0.8809	0.7932
KNN(K=7)	0.8223	0.8254	0.8376	0.8245	0.8335	0.8409	0.8276	0.8345
KNN(K=8)	0.8443	0.8546	0.8576	0.8476	0.8135	0.8265	0.7945	0.8176
KNN(K=9)	**0.8545**	**0.8644**	**0.8665**	**0.8454**	0.8335	0.7689	0.8198	0.8876

Table 5.8: Evaluation of supervised algorithms for Twitter dataset)

Upon analyzing the results in Table 5.7 of the performance metrics for supervised algorithms, it is evident that the Random Forest (RF) model achieves the highest accuracy score when using an 80% training and 20% testing split. The Decision Tree (DT) algorithm

also demonstrates excellent performance, confirming its effectiveness in the task. Logistic Regression (LR), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) algorithms yield acceptable accuracy results. Notably, the KNN algorithm performs exceptionally well when the K value is set to 9, indicating its superiority among the tested K values.

It is important to highlight that these results were obtained after running the algorithms six times, allowing for the selection of the optimal configuration for optimal performance. This underscores the significance of iterative exploration in fine-tuning model results.

Overall, the analysis of Table 5.7 highlights the strengths of different algorithms and emphasizes the importance of thorough experimentation in selecting optimal configurations to achieve high-performance results.

The figure 5.3 presents the bar graph of accuracy for supervised algorithms on twitter dataset.

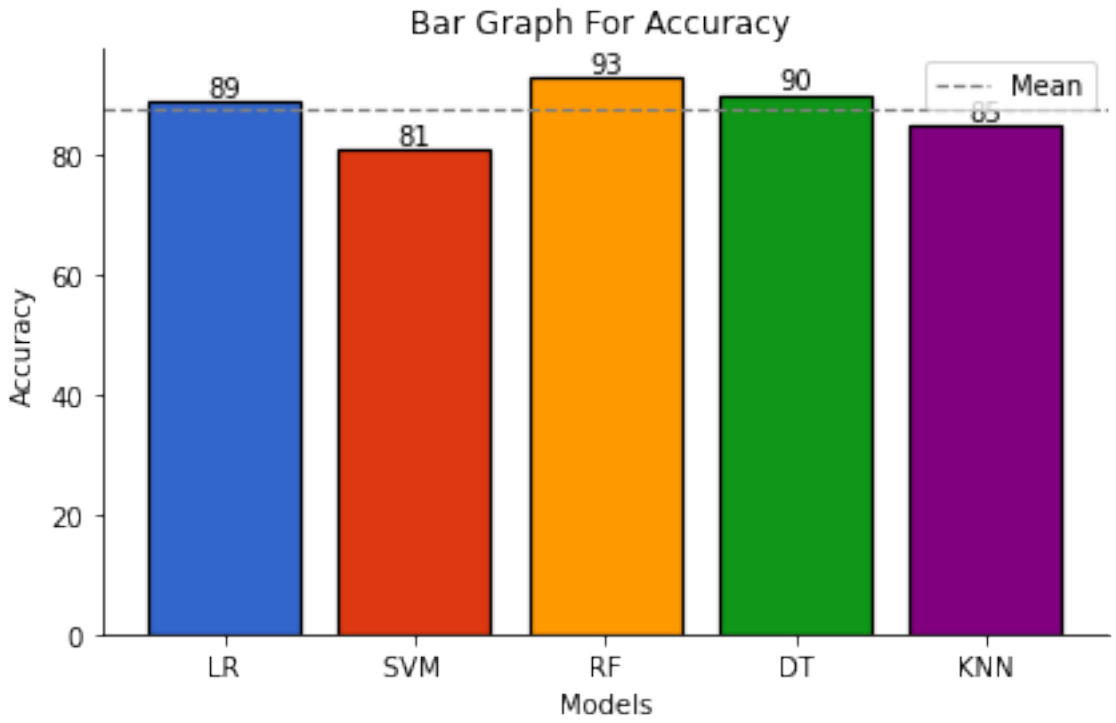


Figure 5.3: *Bar graph for accuracy of supervised algorithms on Twitter dataset*

In Figure 5.3, a bar graph visually depicts the accuracy scores of different supervised algorithms applied to the Twitter dataset. Remarkably, the bar representing the Random Forest (RF) algorithm has the highest value, indicating its superior accuracy performance compared to other algorithms.

2. **Unsupervised Techniques**

The table 5.9 shows the evaluation metrics for both random and manual k-means models on twitter dataset

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	0.2434	0.3545	0.2434	0.3587
Manuel K-means	**0.5098**	**0.7535**	**0.5035**	**0.3454**

Table 5.9: Evaluation of unsupervised algorithm for Twitter dataset

Both the manual k-means and canonical k-means algorithms demonstrate similar performance, with the manual k-means algorithm showing slightly better results. This indicates that the manual approach to k-means clustering may possess specific advantages or optimizations that contribute to its slightly enhanced performance compared to the canonical approach.

Features Selection

In the figure 5.4, the correlation matrix between the features of twitter dataset is shown.

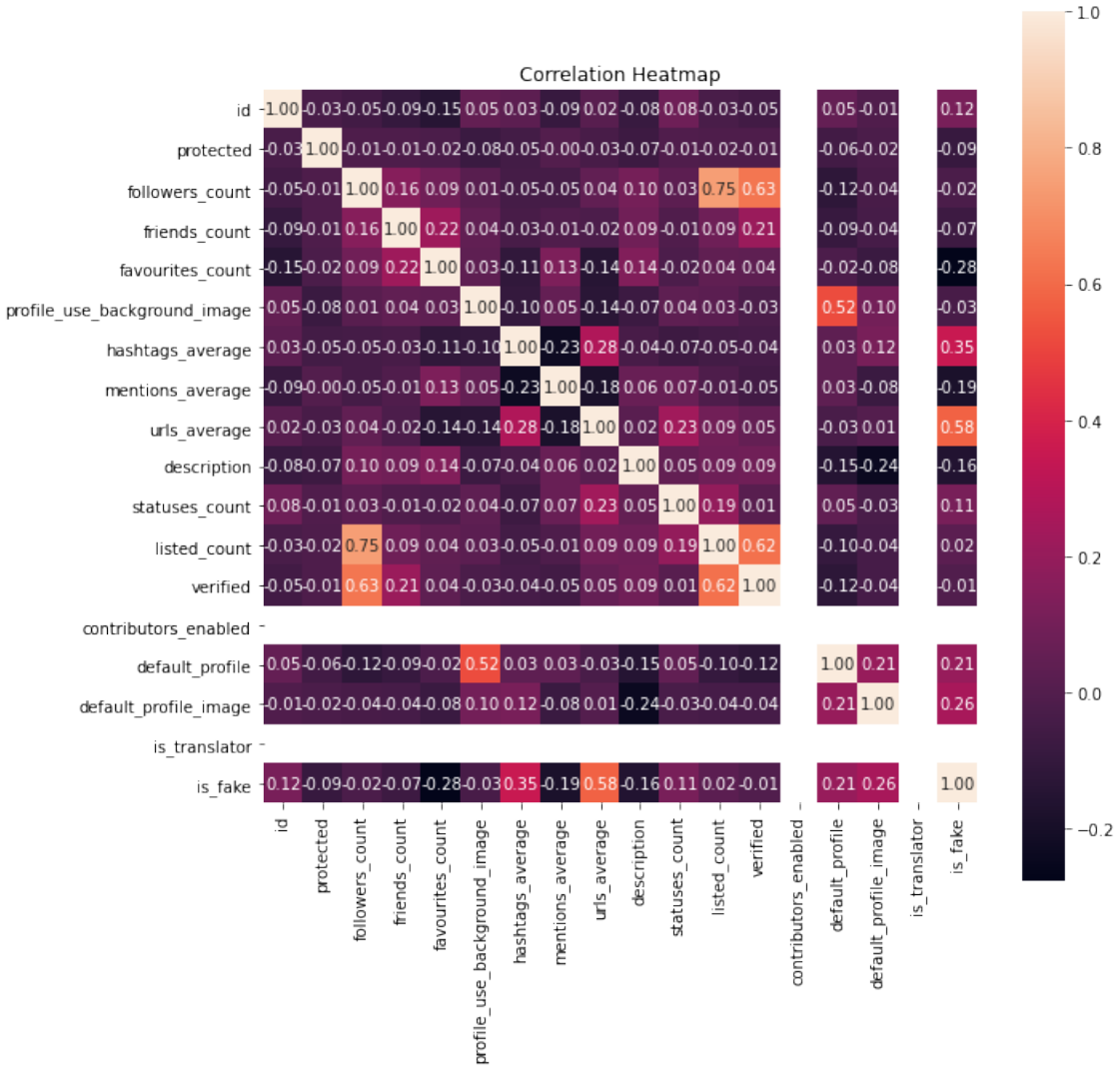


Figure 5.4: *Correlation heatMap of Twitter dataset*

Upon dataset analysis, it is clear that there are insignificant correlations between the features. This indicates that the variables or attributes in the dataset have minimal interdependence or relationship. Consequently, the importance of feature selection diminishes in this scenario.

When there are no significant correlations between features, it implies that each feature contributes unique and independent information. In such cases, removing or selecting specific features may not have a significant impact on the performance of machine learning models or data analysis tasks.

Data normalization

We have scaled(normalized) the twitter dataset in one common range (between 0 and 1) this technique may enhance the models performance .

1. **Supervised Techniques**

   The table 5.10 presents the performance results for supervised models on normalized dataset.

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.9165	0.9187	0.9089	0.9098	0.8554	0.8698	0.8623	0.8543
DT	0.9375	0.9144	0.9367	0.9276	0.8657	0.8776	0.8746	0.8656
SVM	0.8587	0.8756	0.8465	0.8554	0.8565	0.8743	0.8654	0.8576
RF	0.9235	0.9254	0.9157	0.9167	**0.9453**	**0.9257**	**0.9168**	**0.9254**
KNN(K=2)	0.8723	0.8846	0.8865	0.8776	0.8457	0.8587	0.8476	0.8467
KNN(K=3)	0.8645	0.8743	0.8657	0.8555	0.8687	0.8765	0.8654	0.8676
KNN(K=4)	**0.9443**	**0.9456**	**0.9489**	**0.9443**	0.8963	0.9054	0.8935	0.8855
KNN(K=5)	0.8924	0.8965	0.8865	0.8867	0.8524	0.8743	0.8576	0.8646
KNN(K=6)	0.8524	0.8043	0.8557	0.8565	0.8757	0.8835	0.8767	0.8776
KNN(K=7)	0.8624	0.8735	0.8655	0.8643	0.8987	0.8778	0.8887	0.8965
KNN(K=8)	0.8821	0.8865	0.8944	0.8857	0.8655	0.8776	0.8765	0.8654
KNN(K=9)	0.9056	0.9043	0.8978	0.8987	**0.9043**	**0.9054**	**0.8976**	**0.8945**

Table 5.10: Evaluation of supervised algorithms for normalized Twitter dataset)

Random Forests (RF) are insensitive to data scaling as they base decisions on feature values rather than their magnitudes. However, feature scaling (normalization) is crucial

for the K-Nearest Neighbors (KNN) classifier. KNN relies on distance calculations, and if features have varying scales, larger values can dominate the distance calculation and introduce bias into the classification.

Standardizing feature values through normalization significantly improves KNN's performance by ensuring equal importance for each feature and eliminating the influence of different value ranges. Therefore, while RF does not require scaling, standardizing feature values positively affects KNN's classification accuracy.

2. **Unsupervised Techniques** The results of unsupervised techniques on normalized Twitter dataset are shown in Table 5.11 .

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	0.4176	0.4098	0.4054	0.4065
Manuel K-means	**0.5198**	**0.5287**	**0.5187**	**0.4544**

Table 5.11: Evaluation of unsupervised algorithm for normalized Twitter dataset)

Thorough analysis of the data provided in Table 5.11, it is evident that the K-means model's performance becomes satisfactory after employing data normalization techniques. Interestingly, it is worth noting that the manual K-means algorithm mode demonstrates a slight superiority compared to the canonical K-means.

Comparative result on "Twitter dataset" :

The referenced article [8] and this work share the common objective of detecting fraudulent content using machine learning techniques. However, there are notable differences in the methodologies employed in each study. Both works concentrate on detecting fake Twitter profiles using the same dataset. In contrast, there are unique methods and results. The article utilized a supervised discretization technique called Entropy Minimization Discretization (EMD) to preprocess the dataset, specifically focusing on numerical features. The analysis of results was conducted using the Naïve Bayes algorithm. In contrast, we employed a different preprocessing approach for our dataset by utilizing the correlation matrix among the features. Outperformed to conduct feature selection and analysis using various machine learning algorithms such as Random Forest and Decision Tree.

As a result, the Naïve Bayes algorithm yielded an accuracy of 90.41%. This indicates that it outperformed our results obtained on the Twitter dataset.

5.2.3 Instagram dataset

1. **Supervised Techniques**

   The table 5.12 shows the metric for supervised training models on Instagram dataset.

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.9125	0.8976	0.9096	0.8998	0.8798	0.5935	0.5898	0.5867
DT	0.8854	0.8887	0.8865	0.8876	0.8876	0.8856	0.8776	0.8976
SVM	0.9132	0.9245	0.9143	0.9254	**0.9355**	**0.8944**	**0.9054**	**0.9054**
RF	**0.9200**	**0.9245**	**0.9222**	**0.9233**	0.9243	0.9232	0.9324	0.9165
KNN(K=2)	0.8467	0.8535	0.8345	0.8321	0.8732	0.8834	0.8634	0.8754
KNN(K=3)	0.8709	0.8734	0.8765	0.8646	**0.9124**	0.8756	0.9032	0.9146
KNN(K=4)	0.8667	0.8624	0.8543	0.8567	0.9024	0.9043	0.8924	0.9054
KNN(K=5)	**0.8945**	**0.8876**	**0.8912**	**0.8954**	**0.9124**	0.9032	0.9134	0.9043
KNN(K=6)	0.8778	0.8624	0.8745	0.8733	0.9022	0.8945	0.8956	0.9035
KNN(K=7)	0.8998	0.8823	0.8987	0.8856	0.9045	0.9046	0.9165	0.9143
KNN(K=8)	0.8509	0.8756	0.8635	0.8556	0.8932	0.8943	0.9045	0.9043
KNN(K=9)	0.8845	0.8845	0.8921	0.8776	0.8824	0.9032	0.8945	0.8943

Table 5.12: Evaluation of supervised algorithms for Instagram dataset)

Figure 5.12 The experiment results reveal the excellent performance of Random Forest (RF) and Support Vector Machine (SVM) algorithms in the two train-test splits.These findings emphasize the effectiveness of both Random Forest (RF) and Support Vector Machine (SVM) algorithms in various training/testing scenarios. Logistic Regression (LR), K-Nearest Neighbors (KNN), and Decision Tree (DT) algorithms also demonstrate respectable results, albeit not as strong as RF and SVM. These algorithms still showcase their predictive capabilities in the specific context.

Importantly, these results are obtained from multiple runs of the algorithms, ensuring a thorough evaluation of performance. The observed variability across runs adds to the reliability and validity of the findings, acknowledging the inherent fluctuations in algorithm performance.

These findings highlight the strong performance of RF and SVM algorithms, the noteworthy contributions of LR, KNN, and DT, the importance of multiple algorithm runs, and the impact of hyperparameter tuning on KNN. Together, these results contribute to the understanding and advancement of effective algorithmic approaches in machine learning tasks.

The figure 5.5 shows the representation of accuracy for supervised algorithms on Instagram dataset

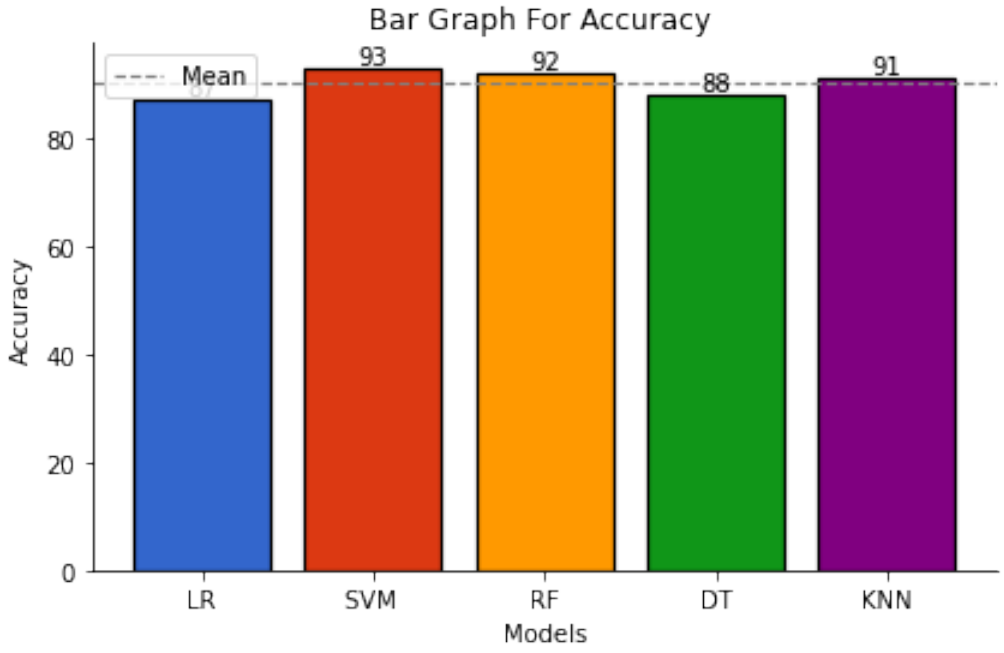


Figure 5.5: *Bar graph for accuracy of different supervised algorithms*

The bar graph presents the accuracy scores of different supervised learning algorithms.

The graph clearly shows that the suport vecteur machine(SVM) algorithm outperforms the other algorithms in terms of accuracy.

The visual representation of the graph serves as concrete evidence, highlighting the superior accuracy performance of SVM. This information is valuable for decision-making purposes, indicating that SVM algorithms are the optimal choice when accuracy is a crucial factor in the predictive model.

2. **Unsupervised Techniques** The performance metrics of canonical and manual k-means in Instagram dataset are shown in the table 5.13 .

valuation of supervised algorithms for Facebook dataset

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	**0.5098**	**0.2543**	**0.5056**	**0.3335**
Manuel K-means	0.5023	0.5034	0.4867	0.1543

Table 5.13: Evaluation of unsupervised algorithms for Instagram dataset)

The evaluation primarily focuses on two main algorithms: manual k-means and canonical k-means. Upon analysis, it becomes evident that both the manual k-means and canonical k-means algorithms exhibit similar and relatively subpar performance.

Features Selection

We find in the figure 5.6 the correlation matrix (heat-map) of Instagram dataset .

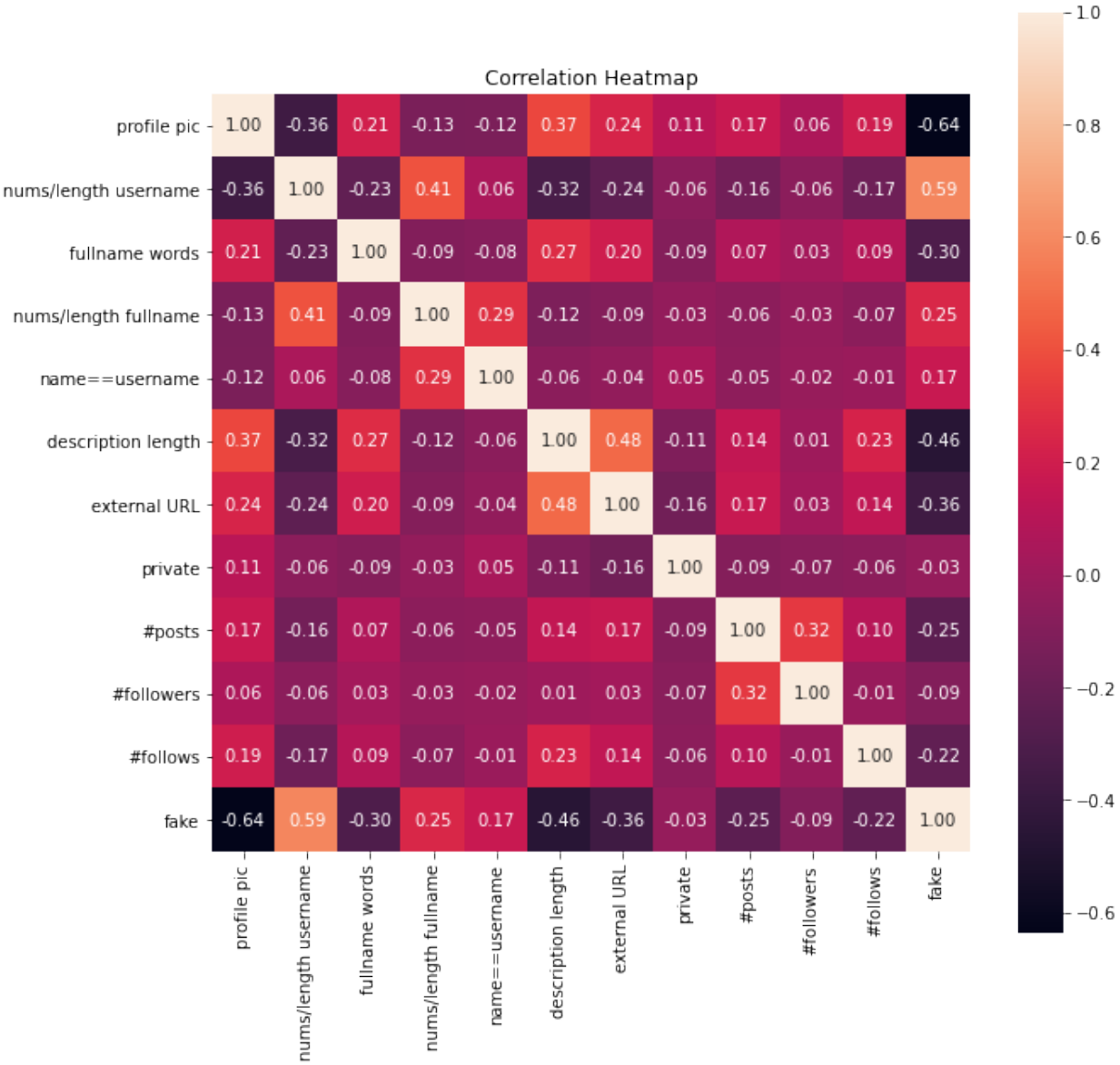


Figure 5.6: *Correlation heatMap of Instagram dataset*

The most features that have dependency on the target class are :

- nums/length username (positive correlation)
- profile pic (negative correlation)
- description length (negative correlation)

- fullname words (negative correlation)
- external URL (negative correlation)

1. **Supervised Techniques**

   The results after features selection on Instagram dataset are shown in table 5.14.

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.8676	0.8656	0.8845	0.8645	**0.9234**	**0.9364**	**0.9234**	**0.9176**
DT	0.8786	0.8934	0.8687	0.8757	0.8545	0.8487	0.9098	0.8578
SVM	**0.8845**	**0.8765**	**0.8787**	**0.8698**	0.8976	0.8987	0.8887	0.8756
RF	0.8756	0.9034	0.8723	0.8699	0.8798	0.8998	0.8787	0.8945
KNN(K=2)	0.8109	0.8655	0.7965	0.7976	0.8698	0.8887	0.8467	0.8576
KNN(K=3)	0.8656	0.8999	0.8598	0.8689	0.8556	0.8698	0.8556	0.8576
KNN(K=4)	0.8478	0.8808	0.8398	0.8376	0.8578	0.8967	0.8467	0.8478
KNN(K=5)	0.8678	0.8956	0.8567	0.8687	0.8687	0.8767	0.8598	0.8487
KNN(K=6)	0.8498	0.8944	0.8387	0.8356	0.8687	0.8956	0.8487	0.8845
KNN(K=7)	0.8656	0.9034	0.8409	0.8576	0.8778	0.8545	0.8476	0.8467
KNN(K=8)	0.8865	0.8323	0.8176	0.8266	0.8587	0.8845	0.8376	0.8365
KNN(K=9)	0.8465	0.8887	0.8287	08276	0.8489	0.8643	0.8366	0.8476

Table 5.14: Evaluation of supervised algorithms for Instagram dataset (After features selection)

Experimental results indicate a slight decrease in the classification accuracy of the test data set when using the proposed feature selection strategy. These results validate the

efficacy of the feature selection strategy in identifying pertinent features that significantly impact the accuracy of predictions and subsequently improve or diminish the performance of classification models.

2. **Unsupervised Techniques**

   The performance metrics for unsupervised techniques in Instagram dataset after features selection are shown in table 5.15.

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	0.3598	0.2409	0.3587	0.2787
Manuel K-means	**0.6287**	**0.7256**	**0.6887**	**0.6187**

Table 5.15: Evaluation of unsupervised algorithm for Intagram dataset (After Features Selection)

Following feature selection on the Instagram dataset, the results of unsupervised algorithms demonstrate notable enhancements in accuracy. Specifically, both canonical k-means and manual k-means algorithms exhibit varying performances. However, it is noteworthy that the manual k-means model surpasses the traditional k-means model, achieving the highest accuracy among them.

Data normalization

1. **Supervised Techniques** The table 5.15 presents the performance results for supervised models on normalized dataset.

Algorithm used	Train – Test 80% - 20%				Train – Test 60% - 40%			
	Accuracy	Precession	Recall	F1-Score	Accuracy	Precession	Recall	F1-Score
LR	0.8798	0.8867	0.8698	0.8754	0.8787	0.8987	0.8834	0.8756
DT	0.8976	0.8965	0.8849	0.8898	0.8779	0.5945	0.5876	0.5864
SVM	0.8786	0.8876	0.8886	0.8643	0.8976	0.9034	0.9134	0.8965
RF	**0.9192**	0.9276	0.9145	0.9145	**0.9474**	0.9465	0.9454	0.9398
KNN(K=2)	0.9092	0.9177	0.9076	0.9098	0.9033	0.9187	0.8798	0.8834
KNN(K=3)	0.9132	0.9078	0.8988	0.8867	0.9098	0.9186	0.9056	0.8734
KNN(K=4)	0.8934	0.9067	0.8998	0.8944	0.9008	0.9287	0.9278	0.9154
KNN(K=5)	0.9013	0.8798	0.8687	0.8954	0.8988	0.8987	0.8887	0.8965
KNN(K=6)	0.9145	0.9278	0.9156	0.9167	0.9097	0.9145	0.9076	0.9045
KNN(K=7)	0.9365	0.9355	0.9354	0.9366	**0.9376**	**0.9334**	**0.9345**	**0.9356**
KNN(K=8)	0.9234	0.8967	0.9276	0.9098	0.8887	0.8977	0.8765	0.8856
KNN(K=9)	**0.9498**	**0.9290**	**0.8989**	**0.9056**	0.8976	0.8999	0.8875	0.8809

Table 5.16: Evaluation of supervised algorithms for normalized Instagram dataset

Decision Trees (DT), Random Forests (RF), and Support Vector Machines (SVM) are not affected by data scaling as they make decisions based on the relative values of features rather than their magnitudes. Therefore, these algorithms are not sensitive to scaling and can perform well regardless of the scaling of the input data. Nevertheless, feature scaling, specifically normalization, plays a crucial role in the K-Nearest Neighbors (KNN) classifier. KNN heavily relies on distance calculations, and when features have disparate

scales, larger values can disproportionately influence the distance calculation, leading to biased classifications. Normalizing feature values through standardization has a significant positive impact on the performance of the K-Nearest Neighbors (KNN) classifier. It ensures equal importance for each feature and eliminates the influence of varying ranges. While Decision Tree (DT) and Random Forest (RF) do not require scaling, standardizing feature values enhances KNN's classification accuracy.

2. **Unsupervised Techniques**

   The performance metrics for unsupervised techniques in Instagram dataset after features selection are shown in table 5.17.

Algorithm used	Accuracy	precision	Recall	F1-Score
Canonical K-means	**0.4456**	**0.4354**	**0.4459**	**0.4368**
Manuel K-means	0.3967	0.2276	0.3987	0.2889

Table 5.17: Evaluation of unsupervised algorithms for normalized Instagram dataset

After conducting a thorough analysis of the data presented in Table 5.17, it becomes evident that the performance of the K-means model becomes satisfactory after applying data normalization techniques. The accuracy value even exhibits a further increase. Interestingly, it is worth noting that the manual K-means algorithm outperforms the canonical K-means model.

Comparative results on "Instagram dataset":

The paper referenced as [5] explores the utilization of machine learning algorithms for the detection and identification of fake accounts on the popular social media platform Instagram.

The authors propose a methodology that involves feature engineering and machine learning algorithms to detect fake accounts on Instagram, utilizing a large dataset. This approach aligns with this work, although we have made some variations in the algorithms used and the resulting outcomes.

The results between our work and the results of the article were varying:

- Both studies utilize different types of regression analysis, specifically linear regression and logistic regression. In both cases, these regression techniques yield satisfactory and convergent results. - It is worth noting that in this work, the Random Forest algorithm outperforms the other regression techniques, resulting in superior outcomes.

Table 5.18 Comparative our work in "Instagram dataset" with another article

	[5]	Our Approche
	Accuracy	
RF	92	92
LR	90	91

Table 5.18: Comparative our work in "Instagram dataset" with another article)

Table 5.18 , a comparison is made between our work on the "instagram dataset" and the

findings presented in the article "etection of Fake Accounts in Instagram using Machine Learning".

The results unequivocally demonstrate that this work attains a higher level of accuracy in comparison to the mentioned article, employing two algorithms: Random Forest (RF) and Linear Regression (LR).

5.3 Bio-Inspired algorithms results

In this section,we evaluate the performance of four bio-inspired algorithms on different previous datasets, to conclusion the algorithm he has the most accurate results .

the results of optimization algorithms are shown in the Tables below.

5.3.1 Facebook dataset

In table 5.19, we find the performance results of bio-inspired algorithms on facebook dataset.

Algorithm used	Accuracy	Precision	Recall	F1-Score
GWO	0.9804	0.9676	0.9723	0.9644
MFO	**0.9959**	**0.9815**	**0.9987**	**0.9843**
GOA	0.9630	0.9287	0.9486	0.9298
WOA	0.9712	0.9809	0.9967	0.9898

The MFO algorithm showed impressive accuracy when compared to other similar algorithms

Table 5.19: Results of four Bio-inspired algorithms for Facebook dataset

in the Facebook dataset. This exceptional performance was consistently observed in six separate runs of the algorithm. Furthermore, the WOA algorithm also exhibited commendable effectiveness in the task. It is important to mention that the GWO and GOA algorithms, which are also inspired by biological systems, achieved reasonably good results but did not reach the same levels as MFO. Nonetheless, they offered valuable insights and potential avenues for future exploration.

5.3.2 Twitter dataset

In table 5.20, we find the performance results of bio-inspired algorithms on twitter dataset.

Algorithm used	Accuracy	Precision	Recall	F1-Score
GWO / LR	0.8150	0.8298	0.8123	0.8112
MFO / LR	0.7770	0.7934	0.7845	0.7734
GOA / LR	0.8220	0.8323	0.8154	0.8113
WOA/LR	**0.8880**	**0.8712**	**0.8888**	**0.8776**

Table 5.20: Results of four Bio-inspired algorithms for twitter dataset

In evaluating biology-inspired algorithms on the Twitter dataset, the WOA (Whale Optimization Algorithm) emerged as the most successful algorithm. This conclusion was reached

after a thorough evaluation involving five separate runs of the algorithm. It is worth noting that the GOA (Grasshopper Optimization Algorithm) also demonstrated remarkable performance, highlighting its effectiveness in tackling the given task. On the other hand, both MFO (Moth Flame Optimization) and GWO (Gray Wolf Optimizer) fell short of achieving the same level of performance as WOA. Nevertheless, despite their relatively lower results, these algorithms provide valuable insights and avenues for further exploration in the field.

5.3.3 Instagram dataset

In table 5.21, we find the performance results of bio-inspired algorithms on twitter dataset.

Algorithms used	Accuracy	Precision	Recall	F1-Score
GWO / LR	0.8278	0.8556	0.8266	0.8124
MFO / LR	**0.9198**	**0.9097**	**0.9109**	**0.9156**
GOA / LR	0.7498	0.7545	0.7334	0.7476
WOA / LR	0.8280	0.8034	0.8123	0.8234

Table 5.21: Results of four Bio-inspired algorithms for instagram dataset

In the analysis of the Instagram dataset, the MFO (Moth Flame Optimization) algorithm stands out for its remarkable accuracy compared to other bioinspired algorithms. These findings were derived from a comprehensive analysis that involved six separate runs of the algorithm.

Additionally, the GWO (Grey Wolf Optimizer) algorithm demonstrated commendable performance, reaffirming its effectiveness in addressing the given task. However, the GOA (Grasshopper Optimization Algorithm) and WOA (Whale Optimization Algorithm) did not achieve the same level of performance as MFO. Despite their comparatively lower results, these algorithms offer valuable insights and potential avenues for further exploration.

5.4 Comparing each machine learning algorithms with bio-inspired algorithms :

5.4.1 Facebook dataset :

In Table 5.20, The comparison between bio-inspired algorithms and machine learning algorithms with the Facebook dataset is depicted.

Algorithms	Accuracy
GWO / LR	0.9804
GOA / LR	0.9630
WOA / LR	0.9712
MFO / LR	**0.9959**
DT	0.9659
RF	**0.9912**
SVM	0.9859
LR	0.9435
KNN(k=2)	0.9061
KNN(k=3)	0.9045
KNN(K=4)	0.8888
KNN(K=5)	0.9247
KNN(K=6)	0.8998
KNN(K=7)	0.8725
KNN(K=8)	0.8888
KNN(K=9)	0.8943

Table 5.22: Bio-Inspired algorithm Vs Machine learning algorithm with Facebook dataset

When comparing bio-inspired algorithms with machine learning algorithms on the Facebook dataset Table 5.22 reveals interesting insights.

It is important to note that the machine learning algorithm RF (Random Forest) achieves the best performance among the machine learning algorithms, while the bio-inspired algorithm MFO (Moth Flame Optimization) exhibits the best performance among the bio-inspired algorithms.

This indicates that RF is highly effective in capturing patterns and making accurate predictions on the Facebook dataset. Similarly, MFO demonstrates remarkable performance, surpassing other bio-inspired algorithms in terms of accuracy. These results underscore the effectiveness of both RF and MFO in their respective domains and highlight their potential for achieving superior performance in data analysis tasks.

Moreover, when considering all the algorithms, including both machine learning and bio-inspired approaches, it becomes clear that MFO (Moth Flame Optimization) and RF (Random Forest) stand out as the top-performing algorithms, achieving an impressive accuracy of 0.99%. This emphasizes the robust predictive capabilities of both MFO and RF in capturing the underlying patterns and generating highly accurate results.

5.4.2 Twitter dataset :

n Table 5.23, The comparison between bio-inspired algorithms and machine learning algorithms with the Twitter dataset is depicted.

Algorithms	Accuracy
GWO / LR	0.8150
GOA / LR	0.7770
MFO / LR	0.8220
WOA / LR	**0.8880**
DT	0.9046
RF	**0.9300**
SVM	0.8132
LR	0.8912
KNN(k=2)	0.7925
KNN(k=3)	0.8132
KNN(K=4)	0.8365
KNN(K=5)	0.8478
KNN(K=6)	0.7656
KNN(K=7)	0.8223
KNN(K=8)	0.8443
KNN(K=9)	0.8545

Table 5.23: Bio-Inspired algorithm Vs Machine learning algorithm with Twitter dataset

In the comparison between bio-inspired algorithms and machine learning algorithms on the Twitter dataset, Table 5.23 provides insightful results. The best performance among machine learning algorithms is achieved by RF (Random Forest), demonstrating its strong predictive capabilities in capturing patterns and making accurate predictions on the Twitter dataset. On the other hand, WOA (Whale Optimization Algorithm) displays the best performance among the bio-inspired algorithms, showcasing its effectiveness in the given task.

Furthermore, when comparing all the algorithms, including both machine learning and bio-inspired approaches, RF (Random Forest) emerges as the top performer, achieving an impressive accuracy score of 0.92. This highlights the superior predictive capabilities of RF and showcases its effectiveness in the given context.

These findings emphasize the significance of evaluating and comparing various algorithmic approaches, encompassing both machine learning and bio-inspired techniques, to determine the optimal solution for a specific dataset. In this context, RF (Random Forest) proves to be the most suitable choice, surpassing other algorithms in terms of accuracy and overall performance. This highlights the importance of selecting the right algorithm for maximizing results in data analysis tasks.

5.4.3 Instgram dataset :

n Table 5.24, The comparison between bio-inspired algorithms and machine learning algorithms with the Instagram dataset is depicted.

Algorithms	Accuracy
GWO / LR	0.8278
GOA / LR	0.7498
WOA / LR	0.8280
MFO / LR	**0.9198**
DT	0.88
RF	**0.9200**
SVM	0.9132
LR	0.9125
KNN(k=2)	0.8467
KNN(k=3)	0.8709
KNN(K=4)	0.8667
KNN(K=5)	0.8945
KNN(K=6)	0.8778
KNN(K=7)	0.8998
KNN(K=8)	0.8509
KNN(K=9)	0.8845

Table 5.24: Bio-Inspired algorithm Vs Machine learning algorithm with Instagram dataset

When comparing bio-inspired algorithms with machine learning algorithms on the Instagram dataset, Table 5.24 provides valuable insights.

The machine learning algorithm RF (Random Forest) achieves the highest performance, indicating its superior predictive capabilities and accuracy in capturing patterns and making precise predictions on the Instagram dataset. In contrast, among the bio-inspired algorithms, MFO (Moth Flame Optimization) displays the best performance, showcasing its effectiveness

in addressing the given task.

Interestingly, it is noteworthy that bio-inspired algorithms (BIAs) generally outperform machine learning algorithms (MLAs) in terms of performance across most cases. This suggests that the nature-inspired techniques implemented in BIAs have a significant impact on enhancing predictive capabilities and overall performance on the Instagram dataset. These results underscore the importance of considering both bio-inspired and machine learning algorithms when tackling complex problems. While RF stands out as the top performer among MLAs, the superiority of MFO among BIAs highlights the potential of bio-inspired approaches in achieving optimal results in the given context. It is essential to explore a diverse range of algorithms to identify the most suitable solution for specific datasets and problem domains.

5.5 Conclusion:

In conclusion, after evaluating machine learning algorithms on three datasets, the Random Forest (RF), Decision Tree (DT), and Support Vector Machine (SVM) algorithms demonstrated strong performance. However, among the bio-inspired algorithms, the Moth Flame Optimization (MFO) algorithm stood out as the top performer, surpassing not only other bio-inspired algorithms but also most of the traditional ML algorithms in terms of precision, accuracy, and recall on the same datasets. This highlights the superiority of MFO within the realm of bio-inspired algorithms and underscores the potential of bio-inspired approaches for achieving optimal results in the given context. Furthermore, the Whale Optimization Algorithm (WOA) demonstrated the second-best performance after MFO in this study, highlighting its potential as an optimization algorithm. However, MFO outperformed WOA across the evaluated metrics.

It is important to note that the comparison conducted in this study was limited to the algorithms and datasets used. Different bio-inspired algorithms may demonstrate strong performance in various scenarios. Therefore, it is crucial to explore a wide range of algorithms to identify the most suitable solution for specific datasets and problem domains.

Chapter 6

Conclusion

6.1 Summary of contributions

A new comparative study

In this dissertation, we conducted a comparative study on fake profile detection techniques using various ONS real-world datasets from Facebook, Twitter, and Instagram. We implemented supervised and unsupervised algorithms, analyzing their performance with different metrics.

It is evident that each algorithm exhibits varying performance in different circumstances, indicating that not all techniques yield consistent results in all environments.

RF, KNN, DT, and SVM are considered the most promising models for the dataset used in this study.

Anomaly detection techniques' results are influenced by the dataset type. Some techniques excel with small datasets but may not be suitable for large datasets. Moreover, certain techniques perform better with sampled and pre-processed data, while others achieve higher accuracies with raw, unsampled data.

Bio inspired algorithms

To enhance the performance of fake profile detection, we acknowledged the limitations of

relying solely on a single machine-learning technique. Hence, we incorporated four bio-inspired algorithms: Grey Wolf Optimizer, Whale Optimization Algorithm, Moth Flame Optimizer, and Grasshopper Optimization Algorithm. By leveraging these diverse algorithms, our aim was to improve the detection capabilities for identifying spam users.

Throughout our research, we achieved noteworthy accomplishments in gaining a deeper understanding of these bio-inspired algorithms. Implementing and analyzing these techniques not only enhanced their performance but also provided insights into their underlying mechanisms. This comprehensive understanding enabled us to assess the strengths and weaknesses of each algorithm and explore their effective application in detecting fake profiles.

The integration of bio-inspired algorithms in this study resulted in significant advancements in the field. It shed light on the potential of these algorithms to enhance the effectiveness of spam user detection, making a valuable contribution to the broader understanding of their efficacy in addressing similar challenges.

6.2 Future works

Future research presents an opportunity for further enhancing this comparative study by incorporating a wider range of bioinspired models. By including and comparing additional bioinspired models, both supervised and unsupervised techniques can be explored to identify the most effective and viable solution. This comprehensive approach would enable researchers to thoroughly evaluate the capabilities and performance of various bioinspired algorithms in addressing the research problem. By considering a diverse set of models, researchers can gain deeper insights into the strengths and weaknesses of each approach, potentially uncovering novel solutions or

improvements. Such efforts would contribute to advancing the field of bioinspired computing and provide valuable guidance for practical applications across multiple domains.

Bibliography

[1] Swati Agarwal and Adithya Samavedhi. Profiling fake news: Learning the semantics and characterisation of misinformation. *BITS Pilani, Goa Campus, Goa, India*, 2022.

[2] XUN YI ZAHIR TARI AHMED ALHARBI, HAI DONG and IBRAHIM KHALIL. Social media identity deception detection: A survey. *ACM Comput. Surv., Vol. 37, No. 4, Article 111. Publication date: January 2021.*, 2022.

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

[4] ALSamuel. Some studies in machine learning using the game of checkers. *ofResearchAnd-Development*, 1959.

[5] Manjistha Dey Niharika Sinha Ananya Dey, Hamsashree Reddy and J Joy. Detection of fake accounts in instagram using machine learning. *Comput. Sci. Inf. Technol*, 2019.

[6] Abdelmajid Ben Hamadou Atika Mbarek , Salma Jamoussi. An across online social networks profile building approach: Application to suicidal ideation detection. *Multimedia InfoRmation Systems and Advanced Computing Laboratory MIRACL, University of Sfax, Tunisia Digital Research Center of Sfax DRCS, Sfax, 3021, Tunisia*, 2022.

[7] Arnab Mitra1 · Anirban Kundu2 · Matangini Chattopadhyay3 · Avishek Bane. An approach to detect fake profiles in social networks using cellular automata-based pagerank validation model involving energy transfer. *The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022*, 2022.

[8] Deniz Kılın¸c Buket Er¸sahin, Ozlem Akta¸s and Ceyhun Akyol. Twitter fake account ¨detection. *In 2017 International Conference on Computer Science and Engineering (UBMK)IEEE*, 2017.

[9] Dave Chaffey. Global social media research summary. *Smart Insights: Social Media Marketing*, 2016.

[10] CNVan de Velde SL Congram, RKPotts. An iterated dyna search algorithm forthe single-machine total weighted tardiness scheduling problem. *on Computing*, 2002.

[11] Ian Davidson. Understanding k-means non-hierarchical clustering. *Computer Science Department of State University of New York (SUNY), Albany*, 2002.

[12] Ananya Dey, Hamsashree Reddy, Manjistha Dey, and Niharika Sinha. Detection of fake accounts in instagram using machine learning. *AIRCC's International Journal of Computer Science and Information Technology*, 11(5):83–90, 2019.

[13] El-GhazaliTalbi. Metaheuristics:from designtoimplementation. 2009.

[14] Mariam Elhussein1. Is it sarrah rahamah? a supervised classification model to detect fake identities on facebook within the sudanese community. *Personal and Ubiquitous Computing*, 2022.

[15] 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.

[16] Nawal Sael Faouzia Benabbou, Hanane Boukhouima. Fake accounts detection system based on bidirectional gated recurrent unit neural network. *International Journal of Electrical and Computer Engineering (IJECE)*, 2022.

[17] Alexander L Fradkov. Early history of machine learning. *IFAC-PapersOnLine*, 53(2):1385–1390, 2020.

[18] Arup Bhattacharjee · Samir Kr. Borgohain · Badal Soni · Gyanendra Verma · Xiao-Zhi Gao. *SMachine Learning, Image Processing, Network Security and Data Sciences*. 2020.

[19] Simran Gibson, Biju Issac, Li Zhang, and Seibu Mary Jacob. Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. *IEEE Access*, 8:187914–187932, 2020.

[20] Simran Gibson, Biju Issac, Li Zhang, and Seibu Mary Jacob. Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. *IEEE Access*, 8:187914–187932, 2020.

[21] Keshav Kaushik1 Akashdeep Bhardwaj1 Manoj Kumar2 Sachin Kumar Gupta3 Abhishek Gupta4. A novelmachine learning-based framework for detecting fake instagram profiles. 2022.

[22] Mark A Hall. *Correlation-based feature selection for machine learning.* PhD thesis, PhD thesis, The University of Waikato, 1999.

[23] Mohammed Azmi Al-Betar Hossam Faris, Ibrahim Aljarah and Seyedali Mirjalili. Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, 2018.

[24] Nuhu Ibrahim and Riza Batista-Navarro. Automatic detection of deaths from social networking sites. *The University of Manchester, Oxford Road, Manchester M13 9PL, UK*, 2022.

[25] Vikramaditya Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.

[26] Joan Daemen Joan Daemen, Vincent Rijmen and Vincent Rijmen. Correlation matrices. the design of rijndael: The advanced encryption standard (aes). 2020.

[27] Nitika Kadam and Sanjeev Kumar Sharma. Social media fake profile detection using data mining technique. *Journal of Advances in Information Technology Vol. 13, No. 5, October 2022*, 2022.

[28] Venu Gopal Kadamba. Evaluation metrics for classification problems with implementation in python. 2021.

[29] Soheyl Khalilpourazari and Saman Khalilpourazary. An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Computing, 23:1699–1722*, 2019.

[30] Rauf Ahmed Shams Malick b Muhammad Sabih a Hocine Cherifi Khubaib Ahmed Qureshi a, . Deception detection on social media: A source-based perspective. *DHA Suffa University, Karachi, Pakistan b National University of Computer and Emerging Sciences, Karachi, Pakistan c University of Burgundy, Dijon, France*, 2022.

[31] Joos Korstanje. The knn model. In *Advanced Forecasting with Python*, pages 169–177. Springer, 2021.

[32] K Leetaru. What does it mean for social media platforms to "sell" our data. *Forbes. https://www. forbes. com/sites/kalevleetaru/2018/12/15/what-does-it-mean-for-social-media-platforms-to-sell-our-data/# 51944f632d6c*, 2018.

[33] Guoyuan Ma and Xiaofeng Yue. An improved whale optimization algorithm based on multilevel threshold image segmentation using the otsu method. engineering applications of artificial intelligence. 2022.

[34] Sri BALA Malladi. Bio inspired computing techniques - an overview. *VRSYRN College of Engineering and Technology*, 2015.

[35] Muhammad Asad Khan Mazhar Javed Awan* and Awais Yasin Zain Khalid Ansari, Hafiz Muhammad Faisal Shehzad. Fake profile recognition using big data analytics in social media platforms. *Int. J. Computer Applications in Technology, Vol. 68, No. 3, 2022*, 2022.

[36] Saman Forouzandeh Mehrdad Rostami. Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 2021.

[37] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. advances in engineering software. 2016.

[38] John Benyen Munga and Prabu Mohandas. Feature selection for identification of fake profiles on facebook. *J. Usman et al. (eds.), 6th Kuala Lumpur International Conference on Biomedical*, 2022.

[39] Umam Mustaqim and Muslim. " application of the nearest neighbor algorithm for classification of online taxibike sentiments in indonesia in the google playstore application".

[40] Mahmoud Fahsi Badia Klouche Nadia Elouali Chourouk Bouhadra Nadir Mahammed, Souad Bennabi. Fake profiles identification on social networks with bio inspired algorithm. 2022.

[41] Indrajit N Trivedi RH Bhesdadiya Pradeep Jangir Narottam Jangir, Mahesh H Pandya and Arvind Kumar. Moth-flame optimization algorithm for solving real challenging constrained engineering optimization problems. *Electronics and Computer Science*, 2016.

[42] JoostNKok EgbertJBoers WalterAKosters PeterVanDerPutten Mannes Poel. Artificialintelligence:definition, trends, techniques, and cases. 2009.

[43] Pratheeksha Hegde N Preethi Salian K Prathyakshini, Nikitha Saurabh. Detection and classification of genui profile based on machine learning techniques. *2022 2nd International Conference on Intelligent Technologies (CONIT) Karnataka, India.*, 2022.

[44] A. Praveena and S. Smys. Effective spam bot detection using glow worm-based generalized regression neural network. *Computer Science and Engineering, RVS Technical Campus, Coimbatore, India*, 2022.

[45] Arnu Pretorius, Surette Bierman, and Sarel J Steel. A meta-analysis of research in random forests for classification. In *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 1–6. IEEE, 2016.

[46] HarishKumar RavneetKaur, SarbjeetSingh. Riseofspamandcompromisedaccountsinonline-socialnetworks. 2018.

[47] T. Anji Reddy A. Saiteja S. Siva Rama Krishna, K. Umakanth Reddy and R. Sumanjali. Detection of fake and clone accounts in twitter using classification and distance measure algorithms. *Intelligent Manufacturing and Energy Sustainability, Smart Innovation, Systems and Technologies*, 2022.

[48] Rohit Satavekar Sahil Mulani, Vani Deshpande Sana Inamdar. Detection of fake and clone accounts in twitter using classification and distance measure algorithms. *GRADIVA REVIEW JOURNAL*, 2022.

[49] Vaman A2019shqi Saeed Saman M. Almufti 1 *, Ridwan Boya Marqas 2. Saman m. almufti 1 *, ridwan boya marqas 2 , vaman ashqi saeed. *Advanced Computer Science Technology*, 2019.

[50] Seyed Mohammad Mirjalili Seyedali Mirjalili and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 2014.

[51] Hossein Shafiei1 and Aresh Dadlani2. Detection of fickle trolls in large-scale online social networks. *School of Engineering and Digital Sciences*, 2022.

[52] Seyedali Mirjalili Shahrzad Saremi and Andrew Lewis. Grasshopper optimisation algorithm: theory and application. *Advances in engineering software,*, 2017.

[53] Seyedali Mirjalili Shahrzad Saremi, Seyedehzahra Mirjalili and Jin Song Dong. Grasshopper optimization algorithm: Theory, literature review, and application in hand posture estimation. 2020.

[54] Dragan Peraković Francisco José García Peñalvo Somya Ranjan Sahoo, Brij B. Gupta and Ivan Cvitić. Spammer detection approaches in online social network (osns): A survey. 2022.

[55] Zoran Stojanovic, Ajantha Dahanayake, et al. *Service-oriented software system engineering: challenges and practices*. Igi Global, 2005.

[56] S Thylashri, U Mahesh Yadav, and T Danush Chowdary. Image segmentation using k-means clustering method for brain tumour detection. *International Journal of Engineering & Technology*, 7(2.19):97–100, 2018.

[57] Using Social Media to Detect Fake News Information Related to Product Marketing: The FakeAds Corpus. Noha alnazzawi 1,* , najlaa alsaedi 2 , fahad alharbi 3 and najla alaswad. *MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations*, 2022.

[58] Putra Wanda. Runmax: fake profile classification using novel nonlinear activation in cnn. 2022.

[59] Putra Wanda and Huang Jin Jie. Deepprofile: Finding fake profile in online social network using dynamic cnn. *Journal of Information Security and Applications*, 52:102465, 2020.

[60] Siti Mariyam Hj. Shamsuddin Zahra Beheshti. A review of population-based meta-heuristic algorithm. *Advance. Soft Comput. Appl., Vol. 5, No. 1*, 2013.