People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
HASSIBA BEN BOUALI UNIVERSITY

THESIS

In preparation for graduation to obtain **software engineer** Diploma

Section : **Computer science**

Speciality : **Software Engineer**

Theme

System for detecting fake users with violent and threatening behavior on social networks.

Presented By :

BENSAIDI Soheyb

BOUZAR ABABOU Djamal

Graduated on : 02/07/2023

Before the juries composed of:

President:	*M* HARBOUCHE Ahmed	Lecturer B
Examiner:	*M* TAYBI Mohammed	Lecturer B
Supervisor:	*M* MAHAMMED Nadir	Lecturer A
Co-Supervisor:	*M* GUELLIL Zouaoui	Teaching Assistant A

Academic year : 2022/2023

Acknowledgements

I am incredibly appreciative to Almighty Allah for giving me the stamina and capacity to do this assignment effectively.

The first thing I want to do is thank our outstanding teacher and supervisor, Mr. Nadir MAHAMMED, as well as Zouaoui GUELLIL, from the bottom of my heart for their unflinching support and direction during the entire process of writing this dissertation. Their guidance has been of immeasurable value to me. Second, I would want to express my profound gratitude to the jury members for their willingness to carefully review, assess, and judge my work. Their knowledge and helpful criticism were crucial in determining how this project will turn out.

I also must take this chance to thank my parents and my complete family from the bottom of my heart. Their unfailing faith in me, coupled with their prayers and unceasing support, has been a source of power that has assisted me in overcoming every challenge along the path.

Last but not least, I would want to express my gratitude to everyone who helped complete this job, whether directly or indirectly. Your participation and encouragement were crucial to its accomplishment.

Dedication

Alhamdulillah! Without the grace and blessings of Allah, we would not have reached where we are today. We humbly dedicate our achievements to the divine hands our parents that guided and protected us, removing the obstacles along our journey and painting our future with lines of hope and confidence. Words of thanks and gratitude are insufficient to express our deep appreciation, Your unwavering support, love, and sacrifices have played an instrumental role in shaping the persons we have become. You have been our guiding light, and we are forever indebted to you. we pray Allah prolongs and blesses ypu're lifes with health and care.

We would also like to express our appreciation to Mr. Mahammed Nadir and Mr. Guellil Zouaoui. Your unwavering support and guidance have been invaluable to us and our journey. You have illuminated our path, providing the necessary encouragement and assistance to overcome any obstacles in our way.

To everyone who has played a part in our life's journey, whether through words of encouragement, acts of kindness, or prayers, we extend our deepest gratitude. Your presence and support have been instrumental in our success, and we are truly blessed to have such wonderful individuals in our lives.

May Allah bless each and every one of you abundantly, and may He continue to guide and protect us on our respective paths.

Résumé

Ces dernières années, l'émergence des plateformes de réseaux sociaux a révolutionné la façon dont les gens se connectent et communiquent. Cependant, ces plateformes sont également devenues des terrains propices à la création et à la propagation de profils frauduleux, ce qui peut avoir de graves conséquences tant pour les individus que pour la société dans son ensemble. La détection et la lutte contre la présence de profils frauduleux revêtent donc une importance capitale. Cette thèse propose une approche novatrice qui combine des techniques d'apprentissage automatique avec un algorithme bio-inspiré appelé optimiseur Fire Hawk (FHO) pour identifier de manière précise les profils frauduleux dans les réseaux sociaux.

La recherche commence par la collecte d'un ensemble de données exhaustif comprenant à la fois des profils authentiques et frauduleux provenant de différentes plateformes de réseaux sociaux. Cet ensemble de données est ensuite prétraité afin d'extraire des caractéristiques pertinentes qui captent les particularités distinctes des profils authentiques et frauduleux. Des algorithmes d'apprentissage automatique tels que les machines à vecteurs de support (SVM) et les forêts aléatoires (RF) sont ensuite entraînés à l'aide de ces caractéristiques pour construire des modèles de classification robustes.

Pour améliorer les performances des modèles de classification, un algorithme bio-inspiré appelé optimiseur Fire Hawk est intégré. FHO s'inspire du comportement de chasse des

faucons et utilise leurs stratégies de chasse pour optimiser le processus de classification. L'algorithme affine itérativement les modèles de classification en ajustant dynamiquement les paramètres du modèle en fonction du paysage de fitness du problème traité.

L'approche proposée est évaluée au moyen d'expériences approfondies sur des ensembles de données réels provenant de réseaux sociaux. Les performances du système sont mesurées en termes de précision, de rappel, de score F1 et d'exactitude. Des comparaisons sont réalisées avec les méthodes existantes de pointe pour démontrer la supériorité de l'approche proposée.

Les résultats montrent que la combinaison de techniques d'apprentissage automatique avec l'optimiseur Fire Hawk permet d'obtenir une précision remarquable dans la détection des profils frauduleux. L'approche présente un potentiel significatif pour des applications réelles, aidant les plateformes de réseaux sociaux à identifier et à supprimer efficacement les profils frauduleux. Cette thèse contribue à l'avancement de la recherche dans la lutte contre les profils frauduleux et fournit une base pour des investigations supplémentaires sur la détection des activités malveillantes sur les réseaux sociaux à l'aide d'algorithmes bio-inspirés.

Mots clé : Faux profile, détection, réseaux sociaux, apprentissage automatique, algorithmes bio-inspirés, simulation.

Abstract

In recent years, the rise of social networking platforms has revolutionized the way people connect and communicate. However, these platforms have also become breeding grounds for the creation and propagation of fake profiles, which can have significant negative consequences for individuals and society as a whole. Detecting and mitigating the presence of fake profiles is thus of utmost importance. This thesis proposes a novel approach that combines machine learning techniques with a bio-inspired algorithm called Fire Hawk Optimizer (FHO) to accurately identify fake profiles in social networks.

The research begins by collecting a comprehensive dataset comprising genuine and fake profiles from various social networking platforms. This dataset is then preprocessed to extract relevant features that capture the distinct characteristics of both genuine and fake profiles. Machine learning algorithms, such as support vector machines (SVM) and random forests (RF)and Gradient Boosting (GB) are trained using these features to build robust classification models.

To enhance the performance of the classification models, a bio-inspired algorithm called Fire Hawk Optimizer is incorporated. FHO is inspired by the hunting behavior of hawks and utilizes their hunting strategies to optimize the classification process. The algorithm iteratively refines the classification models by dynamically adjusting the model's parame-

ters based on the fitness landscape of the problem at hand.

The proposed approach is evaluated using extensive experiments on real-world social network datasets. The performance of the system is measured in terms of accuracy, precision, recall, and F1-score. Comparisons are made with existing state-of-the-art methods to demonstrate the superiority of the proposed approach.

The results show that the combination of machine learning techniques with the Fire Hawk Optimizer achieves remarkable accuracy in detecting fake profiles. The approach exhibits significant potential for real-world applications, helping social networking platforms to identify and remove fake profiles effectively. This thesis contributes to the advancement of research in combating fake profiles and provides a foundation for further investigations into the detection of malicious activities on social networks using bio-inspired algorithms.

Keywords : Fake profiles, detection, social networks, machine learning, bio-inspired algorithms, simulation.

Table des matières

Table des figures

Liste des tableaux

Chapter 1

Introduction

The emergence of social networking websites like Facebook, Twitter and Instagram has fundamentally changed how individuals connect, communicate, and share information on a worldwide scale. These platforms have spawned online communities that enable communication and content sharing among users in various locales. With the advent of online social networking, the dynamics of international communication have undergone a fundamental change, making it possible for people and organizations to interact, have meaningful conversations, and spread information to a large audience like never before. However, these platforms have also seen an increase in the incidence of violent and threatening behavior by some users, in addition to their many advantages. In addition to seriously endangering people's safety and wellbeing, such behavior also runs the risk of upsetting social order and degrading the user experience as a whole. The creation of a reliable method for identifying people who behave violently and threateningly on social media platforms has become essential to resolving this problem. This thesis attempts to propose a comprehensive method for automatically identifying and flagging users engaging in such conduct using cutting-edge methods from natural language processing, machine learning, and data analytics. This will allow for prompt intervention, efficient moderation, and improved user safety on social media.

1.1 Problem statement

Concerns about false accounts (or profiles)' impact on security and privacy on OSNs have grown significantly.

The proliferation of fake profiles on online social media has become a major concern for users and platform providers. These profiles are often created for malicious purposes, such as spreading misinformation, scamming individuals, or conducting cyberattacks.

Detecting fake profiles can be challenging, as they are designed to appear authentic and evade detection. Moreover, traditional methods of identifying fake profiles based on user behavior or profile characteristics are often ineffective, as they fail to capture the dynamic and evolving nature of these profiles. As a result, there is a pressing need for innovative approaches to detecting fake profiles that leverage the power of advanced analytics, machine learning, and artificial intelligence.

1.2 Current literature and motivation

One of the key issues with OSNs is protecting users' security and privacy against false profiles.

The process of identifying bogus accounts has lately been automated and improved by researchers utilizing Machine Learning (ML) techniques. Examples include Bayesian networks or clustering models like K-Means, Naive Bayes, Support Vector Model, and K-Nearest Neighbor.

Researchers have recently done a number of experiments, surveys, and literature reviews to examine and enhance critical thinking ML-based methodologies.

1.3 Contribution and results

In this work, we undertake a novel comparison analysis of various false profile detection strategies in online social networks in order to contribute to enhancing the existing literature. Facebook, Twitter, and Instagram are just a few of the OSNs platforms that are taken into account in our analysis. Additionally, we talk about bio-inspired algorithms.

The goal of our study is not to choose the greatest fake profile detection method out of all those already in use, but rather to choose the approaches within each category that best match the target data sets. The outcomes of our study demonstrate that supervised models

are appropriate for fake profile detection in social media, and that these models can be improved with various machine learning techniques like k-cross validation and parameter tuning, as well as for unsupervised models like k-means and Hierarchical Clustering. The current work suggests a metaheuristic method for locating fake profiles by fusing social media analytics and bio-inspired computing.

1.4 Dissertation structure

this thesis There are four chapters . The background information is provided in Chapter 2, which also places the dissertation in the third chapter of the state-of-the-art. The methods we use to carry out our comparative analysis is presented in Chapter 4. Finally, Chapter 5 gives the results and a discussion of them.

Chapter 2

Basic Concepts

2.1 Introduction

In today's interconnected world, social networks have seamlessly integrated into our daily routines. They serve as indispensable tools for communication, fostering connections, and exchanging ideas. Nonetheless, alongside these remarkable advantages, concerns regarding the misuse of social media are on the rise, notably through the insidious practice of social media manipulation. Fake profiles, meticulously crafted to deceive and exploit unsuspecting users, exemplify this nefarious behavior. These deceptive personas are employed for various malicious purposes such as spamming, phishing, disseminating deceptive information, potentially leading to identity theft, and even perpetuating cyberbullying. It is imperative that we remain vigilant, recognizing the presence of these fake profiles, and working towards a safer and more authentic digital landscape. This chapter includes a review of various methods. But first, now let us clarify the fundamental concepts surrounding this subject.

2.2 Fake profiles detection

2.2.1 Web 2.0

is a term used to describe a new generation of web-based technologies and platforms that facilitate user-generated content, collaboration, and interaction. It represents a shift from static, one-way communication to dynamic, interactive communication. Web 2.0 applications include social networking sites, blogs, wikis, and other tools that allow users

to create and share content, participate in online communities, and collaborate on projects in real-time. This has transformed the internet into a more collaborative and participatory space [35].

2.2.2 Online Social Networks

are virtual platforms that allow people to create and maintain social connections, share content, and communicate with others over the internet. Social networks have become an integral part of modern life, with billions of users around the world connecting and sharing information through platforms like Facebook, Twitter, and LinkedIn. These platforms have transformed the way people interact and communicate, enabling individuals and groups to connect with each other regardless of geographic location.[2]

One of the key features of online social networks is the ability to create personal or professional profiles. Users can create a profile that includes information such as their name, location, interests, and a profile picture. They can also connect with other users by sending friend requests, following other users' profiles, and joining groups or communities based on shared interests. Social networks also allow users to share content such as photos, videos, and messages with their connections. This has facilitated the creation of new forms of digital media, including user-generated content and online communities. The rise of online social networks has had a profound impact on society, transforming the way people communicate and interact with each other. Social networks have facilitated the exchange of information and ideas, giving rise to new forms of digital media and online communities. They have also transformed the way businesses and organizations communicate with their audiences, enabling more targeted and personalized marketing and advertising. Despite the benefits of social networks, however, they have also raised concerns about privacy, security, and the potential for misinformation and online harassment.Due to the ease with which personally identifiable information, such as age, gender, full name, and address, can be misused, many OSN users are ignorant of the security dangers associated with these kinds of communications, including the possibility of fake profiles.

2.2.3 Fake profile problem

Fake profile :An account with fake credentials in the name of a celebrity or any person, organization or company, it can be believed as real and used for advertisements and marketing to give misleading information to the followers. Fake profiles are often created

with the intention of spreading misinformation or engaging in fraudulent activities, such as scamming or phishing. Identifying and removing these profiles is crucial to maintain the integrity and trustworthiness of social media platforms [36] . 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.

Fake profiles can be used to carry out malicious activities such as harming person's reputation and privacy in OSN. They can be considered as one of the distinctive attacks in online social media.

Fake profiles types

Several Fake Profiles types can be identified including:

- compromised profiles.
- clone profiles.
- spam bots

2.3 Machine Learning

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and techniques to enable computers to learn from data and make predictions or decisions without being explicitly programmed. The fundamental principle behind machine learning is the ability of algorithms to automatically learn patterns and relationships within data, leading to the creation of models that can make accurate predictions or take informed actions. This data-driven approach has revolutionized various industries, from healthcare and finance to marketing and transportation, enabling businesses to gain valuable insights and improve decision-making processes.

2.3.1 Machine learning models

There are multiple possible setups when following an ML-based AD. On one hand, the setup to be used may depend on the availability of labels indicating the actual nature of the initial training data.[37]

- **Supervised learning** :Supervised learning is one of the core branches of machine learning. In supervised learning, a model is trained on labeled data, where each data point is associated with a known target variable or outcome. The goal is to learn a mapping between the input features and the corresponding output labels. Classification and regression are common tasks in supervised learning. Classification involves predicting a discrete class or category, while regression aims to predict a continuous value. Algorithms such as decision trees, support vector machines, and neural networks are commonly employed in supervised learning.. KNN, DT, SVM, LR, Artificial Neural Network (ANN), Naive Bayes (NB) etc., are some popular algorithm of supervised learning.

- **Unsupervised learning**: Unsupervised learning, in contrast to supervised learning, deals with unlabeled data, where the model learns patterns or structures in the data without specific output labels. Clustering and dimensionality reduction are common unsupervised learning tasks. Clustering algorithms group similar data points together based on their inherent similarities or distances, revealing hidden structures in the data. Dimensionality reduction techniques aim to reduce the complexity of high-dimensional data while preserving important information by projecting it into a lower-dimensional space. Principal Component Analysis (PCA) and k-means clustering are widely used unsupervised learning algorithms.

Supervised models

Naive bayes classifier Naive bayes is a machine learning algorithm based on bayes theorem it is a supervised learning task and also assumed that the predictive attributes are independent. We define the Bayes theorem as :

$$P(class/features) = P(class) * P(features/class)/P(features) \tag{2.1}$$

- P(class/features) : Posterior Probability

- P(class) : Class Prior Probability
- P(features/class) : Likelihood
- P(features) : Predictor Prior Probability

Bayesian classifier is based on Bayes' theorem. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered "naive".[1]

Bayesian classifier is based on Bayes' theorem. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered "naive".

Remarks on the Naive Bayesian Classifier (figure 2.1) :

- Studies comparing classification algorithms have found that the naive Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers.
- Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases [1].

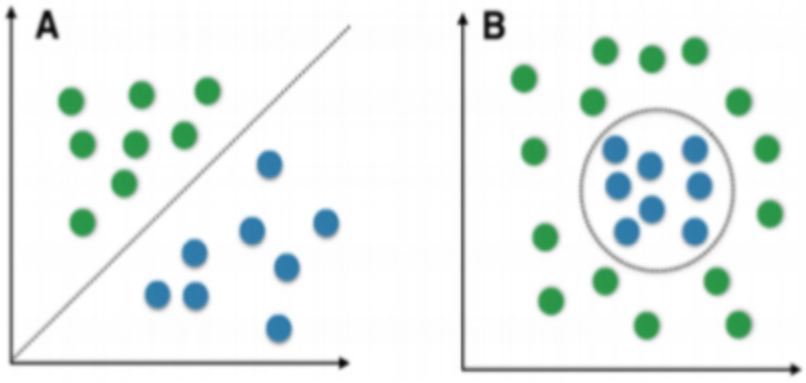

Figure 2.1: *Naive Bayes classifier [1]*

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 2.2 [1])

The idea is to partition the data space into dense regions and sparse regions. It is a statistical-based algorithm where attributes are selected at the tree-of-nodes beginning at the root and ending at the leaves.

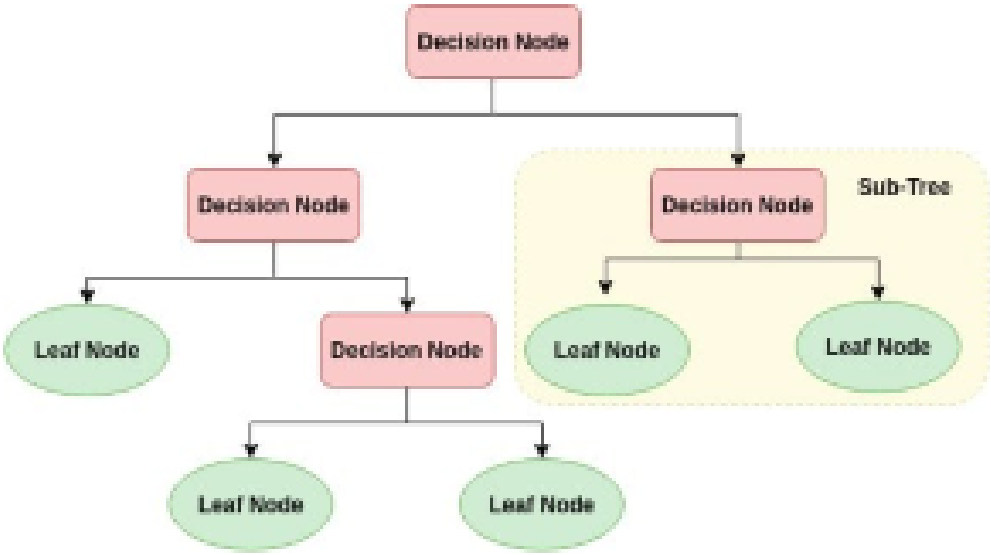

Figure 2.2: representation of decision tree [2]

K-nearest neighbors 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 [2].

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.[2]

One of the drawbacks of K-Nearest Neighbors is that it is sensitive to inconsistent data

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

(noisy) and missing value data. The figure below shows how the knn classifier works.

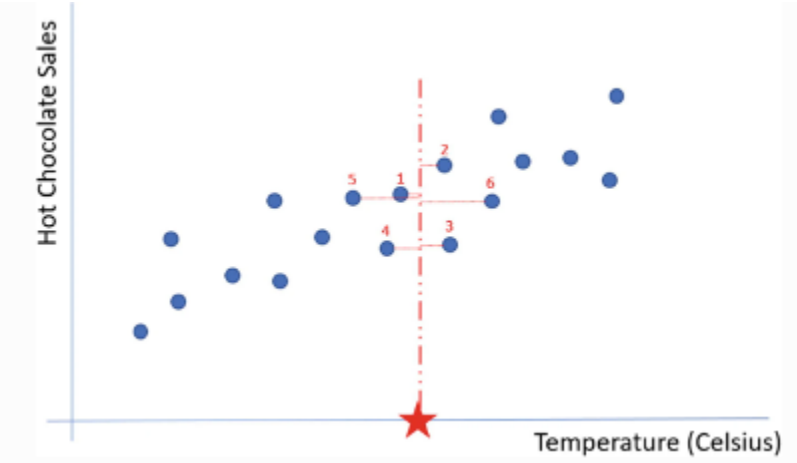

Figure 2.3: KNN classifier [2]

GRADIANT BOOSTING Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[3] [38] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.[3] [38] [39] A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

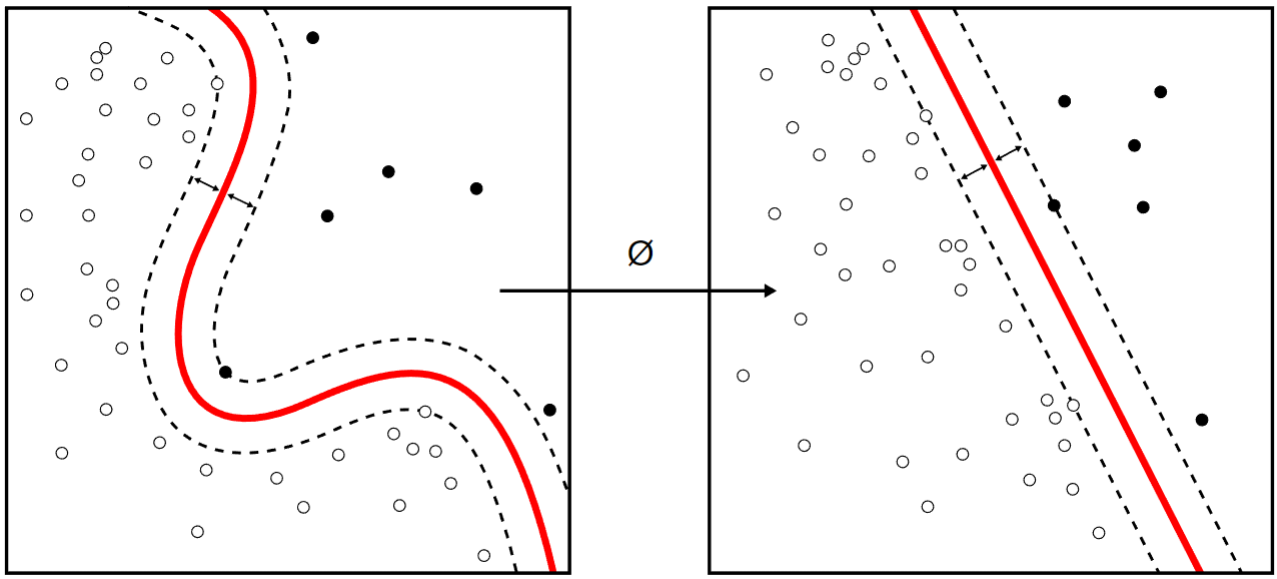

Figure 2.4: GRADIANT BOOSTING[3]

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)[40]

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.[4]

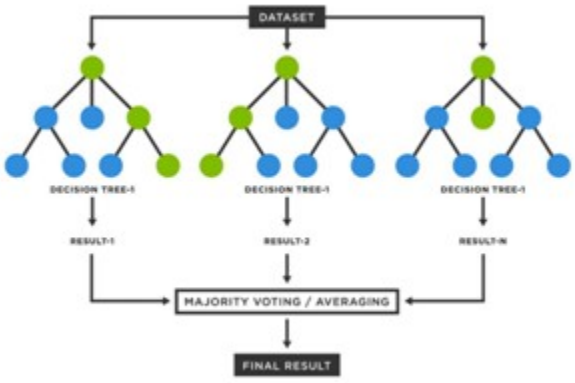

Figure 2.5: Random Forest Classifier [4]

Support Vectors Machine Support Vector Machines (SVMs) are supervised learning models for classification and regression problems. Support Vector Machines(SVMs) are supervised learning models for classification and regression problems. They can solve linear and nonlinear problems and use the concept of Margin to classify between 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 [41].

- Support Vectors are the data points that are on or closest to the hyperplane and influence the hyperplane's position and orientation.
- Hyperplanes are decision boundaries that aid in classifying the data points.

  The figure 2.6 shows the representation pf hyper planes

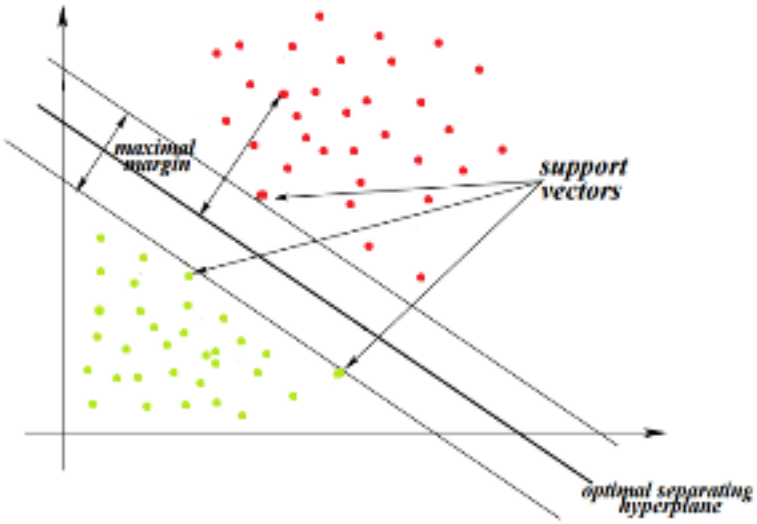

Figure 2.6: SVM Classifier [5]

Unsupervised model

Kmeans algorithm K-means clustering is one of the simplest and popular unsupervised machine learning algorithms .

For an attribute problem, each instance maps into a m dimensional space. The cluster centroid describes the cluster and is a point in m dimensional space around which instances belonging to the cluster occur.

The distance from an instance to a cluster center is typically 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 [42].

- A cluster refers to a collection of data points aggregated together .
- K is a target number, which refers to the number of centroids you need in the dataset.
- A centroid is the imaginary or real location representing the center of the cluster.

k-means clustering tries to find the similarity between the items and groups them into the clusters. K-means clustering algorithm works in three main steps.

- Select the k values.
- Initialize the centroids.

- Select the group and find the average.

The figure 2.7 shows the flowchart of kmeans clustering

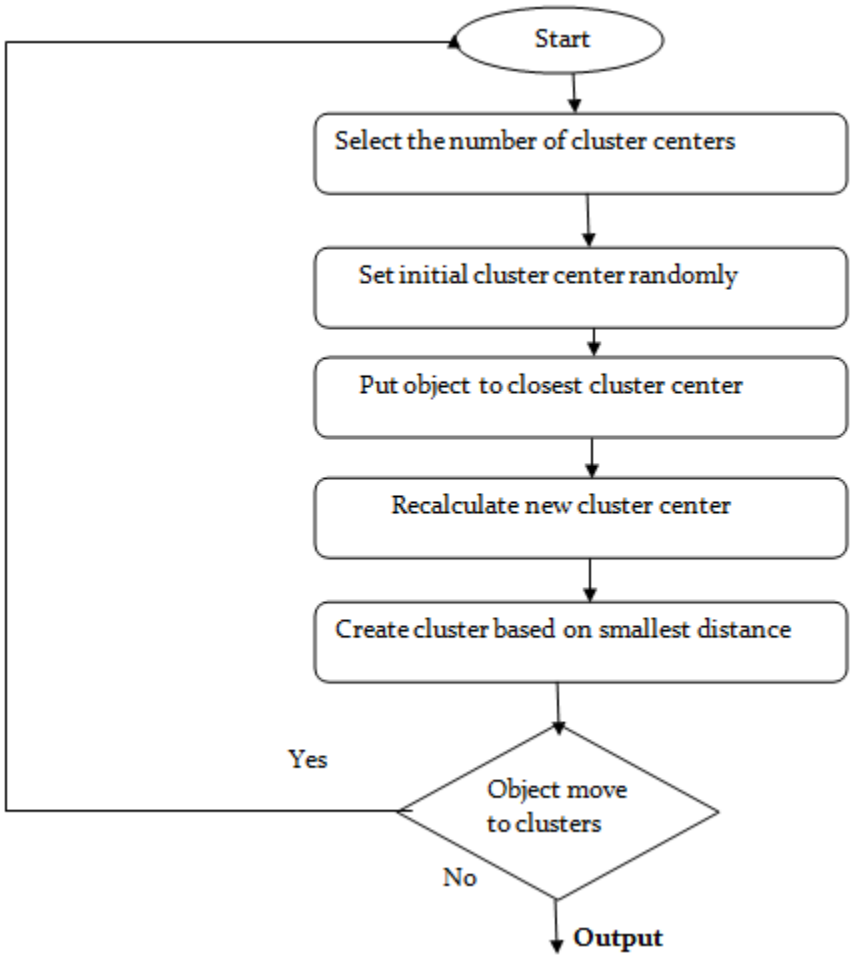

Figure 2.7: Flowchart of k-means clustering algorithm [6]

Hirarchical clustering Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. An endpoint is a collection of clusters, where each block is different from each other, and the objects within each block are very similar to each other.

Hierarchical grouping begins with treating each note as a separate group. Then, it recursively performs the following two steps:

- identifying the two groups that are closest together.

- merging the two groups that are most similar.

This iterative process continues until all groups have been merged.

Semi-Supervised Machine Learning:

Problems where we have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning problems.These problems sit in between both supervised and unsupervised learning.

Example photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled where the unsupervised learning techniques used to make best guess predictions for the unlabeled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data.

2.4 Metaheuristic

Metaheuristics are techniques for solving problems that effectively search across a sizable solution space in search of ideal or nearly ideal answers. These tactics are used to direct the search process away from local optima and toward promising areas of the solution space. Many optimization issues, such as combinatorial, continuous, and multi-objective optimization, can be solved using metaheuristics. Genetic algorithms, simulated annealing, ant colony optimization, particle swarm optimization, and tabu search are a few examples of metaheuristic algorithms. When a problem needs a substantial amount of computer resources or is too difficult to be addressed using conventional optimization techniques, metaheuristics are frequently applied [43].

2.4.1 Metaheuristics and machine learning

In recent years, there has been growing interest in combining metaheuristics with machine learning techniques to further improve their efficiency and effectiveness in solving complex optimization problems. By integrating machine learning techniques into metaheuristics, it is possible to automate the parameter tuning process, improve the decision-making process, and learn heuristics from past solutions[7].

The success of machine learning-based metaheuristics has been demonstrated in various optimization problems, including hyperparameter optimization, feature selection, and

portfolio optimization, among others. Deep reinforcement learning-based metaheuristics, neural network-based metaheuristics, and evolutionary algorithms with machine learning components are some examples of machine learning-based metaheuristics that have shown promising results in solving optimization problems.

Machine learning techniques are also widely used in various fields, such as search engines, robotics, computer vision, finance, bioinformatics, and insurance, among others. With the increasing availability of data and computing resources, machine learning-based metaheuristics are expected to become even more powerful and effective in solving complex optimization problems in the future.

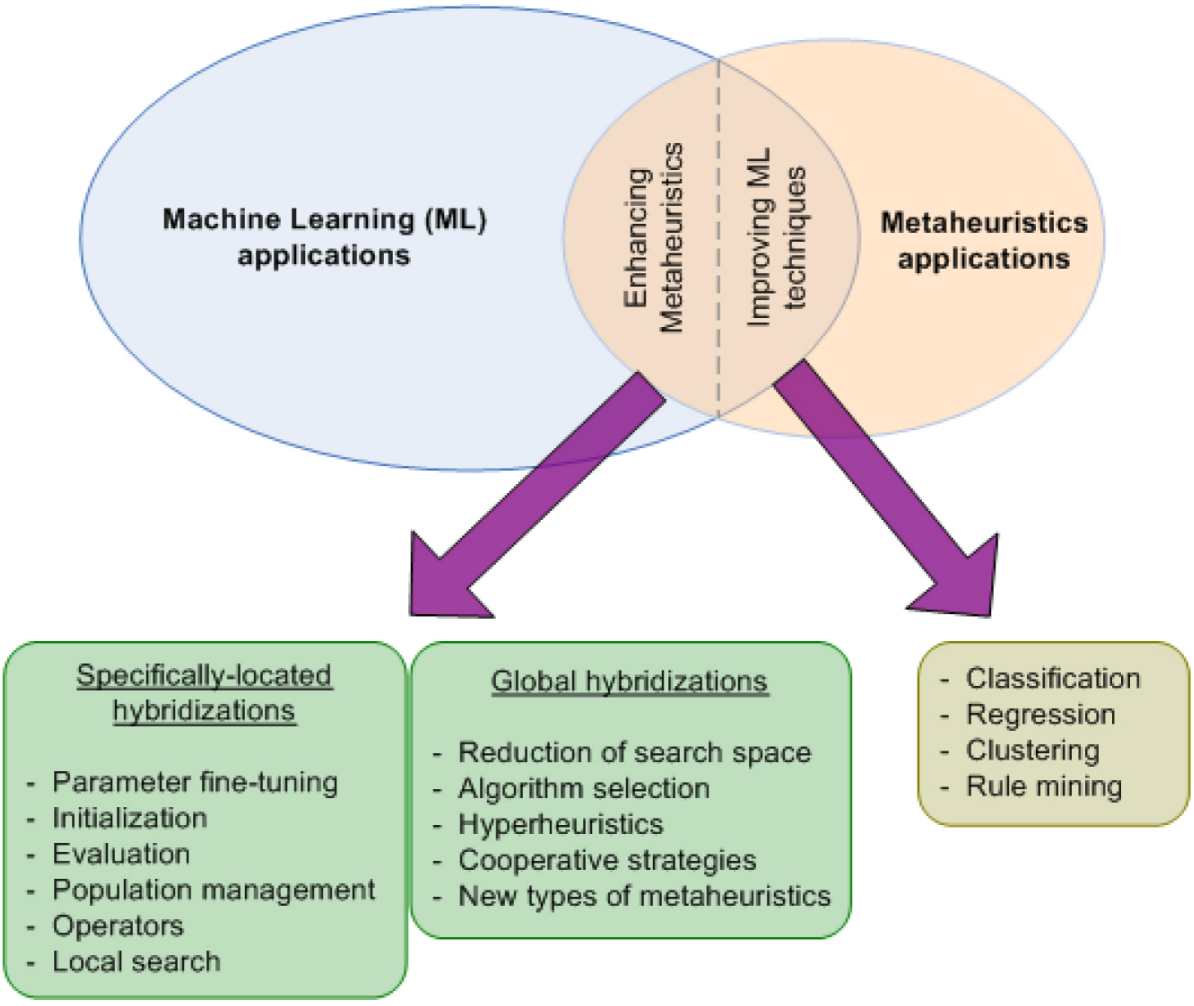

Figure 2.8: ML and Metaheuristics[7]

2.4.2 A Brief History

The history of metaheuristics as scientific methods for optimization can be traced back to the mid-20th century when heuristic approaches were used in various applications. The development of evolutionary algorithms marked a significant milestone, with Ingo Rechenberg and Hans-Paul Schwefel introducing evolutionary strategies in 1963 and L. J. Fogel et al. proposing evolutionary programming in 1966. J. Holland later pioneered genetic algorithms in the 1960s and 1970s, publishing a seminal book on the subject in 1975.[44]

The 1980s and 1990s witnessed notable advancements in metaheuristic algorithms. Simulated annealing, inspired by the annealing process of metals, was introduced by S. Kirkpatrick et al. in 1983. The development of artificial immune systems by Farmer et al. in 1986 was another significant step. Glover pioneered the use of memory in metaheuristics with Tabu search in the 1980s, publishing a comprehensive book on the topic in 1997.

In 1992, Marco Dorigo presented his innovative work on ant colony optimization (ACO) in his PhD thesis. Genetic programming, introduced by John R. Koza in 1992, revolutionized computer programming. Particle swarm optimization was developed by James Kennedy and Russell C. Eberhart in 1995, followed by the vector-based differential evolution algorithm by R. Storn and K. Price in 1996-1997.[45]

Entering the 21st century, more exciting developments unfolded. The harmony search algorithm by Zong Woo Geem et al. in 2001 drew inspiration from music. Bacteria foraging and honey bee algorithms emerged in 2002 and 2004 respectively, with subsequent variations like the artificial bee colony algorithm. In 2008, the firefly algorithm was introduced, while 2009 saw the efficient cuckoo search algorithm by Xin-She Yang and Suash Deb, outperforming existing metaheuristics including particle swarm optimization [46] [47][48].

2.4.3 mode of operation of metaheuristics

Metaheuristics operate according to a set of standard guidelines and procedures. An initial population or solution set is created first, frequently randomly or with the use of heuristics. This population represents potential answers to the current optimization issue. After that, an iterative process starts, with each iteration denoted by the terms generation or iteration. The metaheuristic algorithm assesses the fitness or quality of each population solution

within each generation. The standard method for evaluating fitness is to use an objective function or a set of constraints that are unique to the optimization issue. Higher fitness values are regarded as superior or being nearer to the ideal solution. The method uses a variety of operators or transformations to provide new candidate solutions after evaluating fitness. These operators might use local search methods, mutation, or recombination. The goal is to explore the search space for new, possibly superior solutions.

The algorithm then chooses a subset of solutions to create the next set of solutions, frequently based on their fitness levels. Elitism, which preserves the best solutions, or stochastic selection techniques, like roulette wheel selection or tournament selection, may be used in this selection process.

Up until a termination condition is satisfied, the iteration process keeps going. The number of generations, reaching a particular fitness threshold, or going over a computational time restriction can all be used as termination criteria.

Throughout the iterations, the metaheuristic algorithm maintains a memory or history of previously evaluated solutions. This memory can help guide the search process, avoiding revisiting already explored solutions or promoting diversification and intensification.[49][50]

Overall , Metaheuristics generally operate by creating an initial population, iteratively assessing fitness, applying operators to create new solutions, choosing the best solutions for the following generation, and continuing the process until a termination condition is satisfied. Memory enhancement and successful search space navigation are two benefits of memory use.[49]

2.4.4 Classification

Local search and global search

Defining the sort of search strategy is one strategy. An improvement over straightforward local search algorithms is one kind of search method. The hill climbing algorithm, which is used to locate local optimums, is a well-known local search technique. Hill climbing does not, however, ensure that the world's best solutions will be discovered.

To enhance local search heuristic and locate better solutions, many metaheuristic theories were put forth. Simulated annealing, tabu search, iterated local search, variable neighborhood search, and GRASP are some examples of these metaheuristics. Both of these metaheuristics fall within the local search-based and global search categories.

Population-based metaheuristics are typically used for other global search metaheuristics that are not based on local search. Ant colony optimization, evolutionary computation, particle swarm optimization, genetic algorithm, rider optimization method, and others are examples of such metaheuristics.[51][52][53]

Single-solution vs. population-based

Metaheuristics can be categorized as single-solution or population-based. Single-solution approaches manipulate a single candidate solution iteratively, while population-based approaches maintain a population of solutions, allowing for exploration and exploitation. Population-based metaheuristics often exhibit better search capabilities but require additional computational resources.[51][54][55]

Hybridization and memetic algorithms

ybridization refers to the combination of different metaheuristic algorithms or techniques to create a more powerful and effective optimization approach. It leverages the strengths of multiple algorithms to enhance exploration and exploitation capabilities, ultimately improving the quality of solutions obtained. Memetic algorithms are a specific type of hybrid metaheuristics that combine global search strategies with local improvement heuristics to enhance solution quality and convergence speed.[56]

Parallel metaheuristics

Hybridization is the process of combining various metaheuristic algorithms or methodologies to produce a more potent and successful optimization strategy. It makes use of the advantages of several algorithms to improve the capabilities of exploration and exploitation, finally raising the caliber of solutions produced. Memetic algorithms are a particular class of hybrid metaheuristics that accelerate convergence and increase solution quality by fusing local improvement heuristics with global search techniques.

Nature-inspired and metaphor-based metaheuristics

Natural events or processes are used as inspiration for nature-inspired metaheuristics, which direct the search for the best answers. Examples include ant colony optimization, particle swarm optimization, and genetic algorithms. These algorithms use ideas like

evolution, swarm behavior, and pheromone communication to simulate biological, physical, or ecological systems.

On the other hand, metaphor-based metaheuristics use metaphors from other areas to create optimization algorithms. They use concepts and tenets from domains unrelated to their own, including social behavior, cultural development, or artistic processes. Metaphor-based metaheuristics provide novel approaches in optimization and the exploration of non-conventional solution spaces by providing distinctive viewpoints and creative problem-solving techniques.[8]

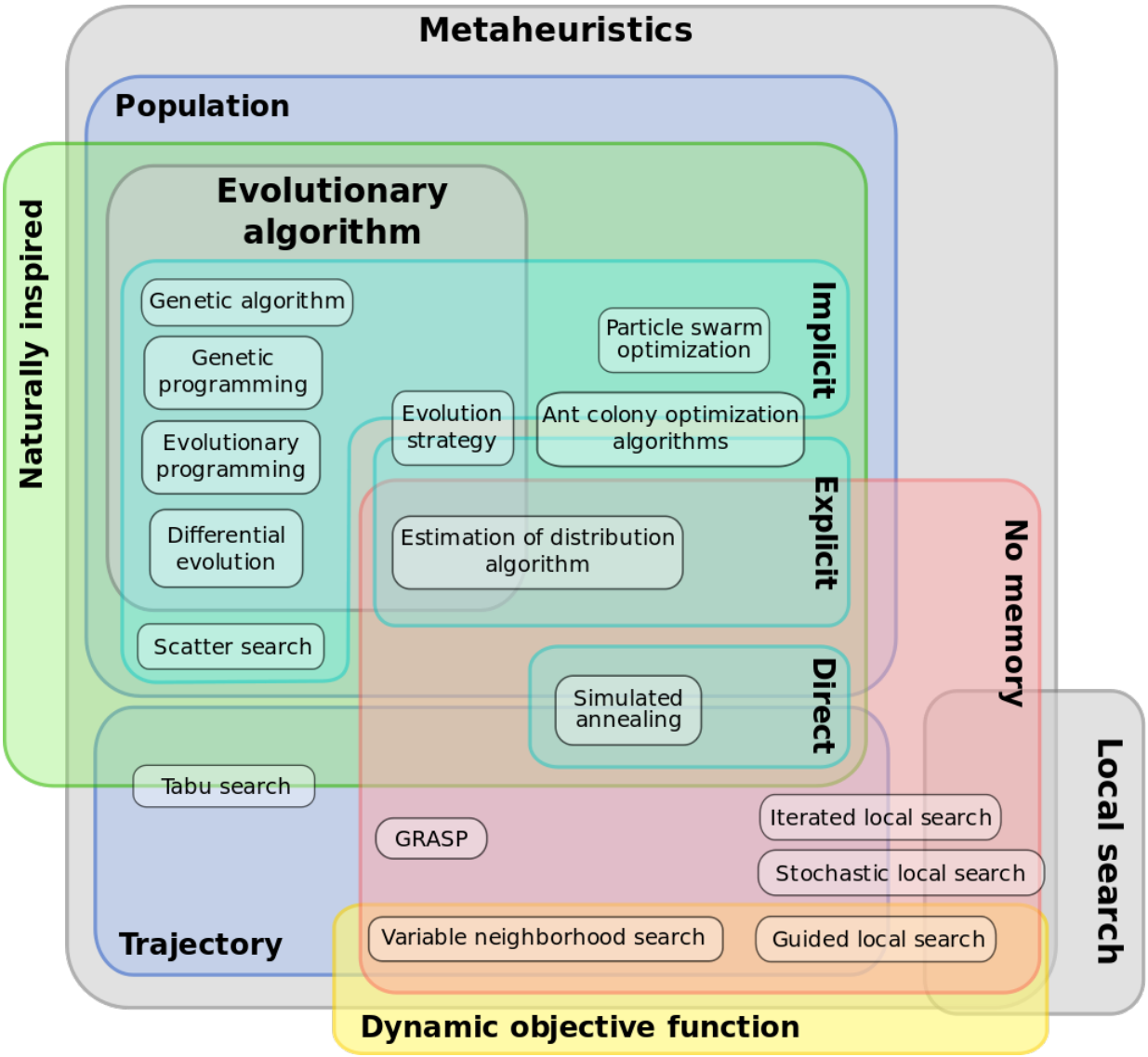

Figure 2.9: Euler diagram of the different classifications of metaheuristics[8]

2.4.5 Using machine learning for enhancing metaheuristics

The study of how machine learning methods have been applied to improve metaheuristics has been split into two sections for the sake of clarity. The first portion examines local-level hybridizations, while the second section examines global-level hybridizations. Each of them has been categorized by the appropriate topic [43].

2.5 Bio-inspired algorithms

Bio-inspired algorithms, are a class of optimization algorithms that draw inspiration from natural processes and phenomena. These algorithms are designed to solve complex optimization problems that are often difficult to solve using traditional mathematical methods.

Some examples of bio-inspired algorithms include genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), differential evolution (DE), and artificial immune systems (AIS).

GA is based on the idea of natural selection and genetics, where solutions are represented as chromosomes and evolve through selection, crossover, and mutation. PSO is inspired by the behavior of swarms and flocks, where solutions are represented as particles that move in a multidimensional search space to find the optimal solution. ACO is inspired by the behavior of ant colonies, where agents deposit pheromones to communicate and find the shortest path between a food source and their nest. DE is based on the idea of natural selection and evolution, where a population of candidate solutions evolve through differential mutation and crossover operations. AIS is inspired by the human immune system, where solutions are represented as antigens and undergo a selection process to identify and eliminate anomalies.

Bio-inspired algorithms have been applied in various fields such as engineering, finance, biology, and computer science. These algorithms have shown promising results and are becoming increasingly popular due to their ability to efficiently solve complex optimization problems[57].

2.5.1 Classification of BIAs

BIAs are still in the stage of development, so there is no strict definition and uniform classification. Binitha and Sathya described the origin and advantages of the bioinspired computing algorithms and pointed out that BIAs were heuristic methods that imitated the strategy of nature, which was a simple and nonrepresentational definition of BIAs. Bongard introduced the development process of the bioinspired computing algorithm and analyzed the relationship between the BIAs and the traditional intelligent computing methods. Based on our expertise, the bioinspired intelligent computing algorithm can be defined as follows: it is a type of intelligent computing methods with a quite lifelike biological working mechanism, to imitate the function and structure of the organism, the individual and swarm behaviors, and the evolution process of life and society. However, it is not an easy job to exclude those methods from BIAs, which are not strictly bioinspired. In this paper, we do not intend to distinguish different types of intelligent computing algorithms but to analyze the main features of BIAs, classify these algorithms from the simulated biological working mechanisms, and survey different categories focused on the realization processing as well as the applications for mobile robot control[9].

Based on our research work and the overview of lots of literature, the remarkable features of BIAs discussed are summarized as follows :

- (1) Bioinspired feature: the working mechanism is very close to the biological or ecological mechanism of natural organisms. The bioinspired algorithms imitate the biological nature as much as possible to deal with the real-world problems.

- (2) Simplicity and emergence: the strategy and computation are often very simple, but their resultant effects are very amazing, which reflects the principle of emergence.

- (3) Robustness: these algorithms have strong robustness against the change of environments, parameters, and tasks; namely, these algorithms have good applicability and flexibility.

- (4) Self-organization: these algorithms can improve the adaptability by self-learning or self-organization and realize the evolution successfully.

- (5) Other features: these algorithms have some other good features, such as parallelism in essence and nondeterminism.

BIAs have many excellent characteristics that can meet the needs of researchers. To introduce BIAs clearly and understand them easily, BIAs should be classified. From different view angle, various classifications can be obtained; for example, from the whole effects, we can see all BIAs as a type of evolutional optimization algorithms.

Bioinspired intelligent algorithms (BIAs) imitate lifelike biological working mechanisms, making them promising solutions. BIAs are heuristic methods that imitate nature's strategy and are classified based on the biomimetic mechanism, including behavior, structure, and evolution. BIAs possess several characteristics, including bioinspired features, simplicity and emergence, robustness, self-organization, and parallelism.

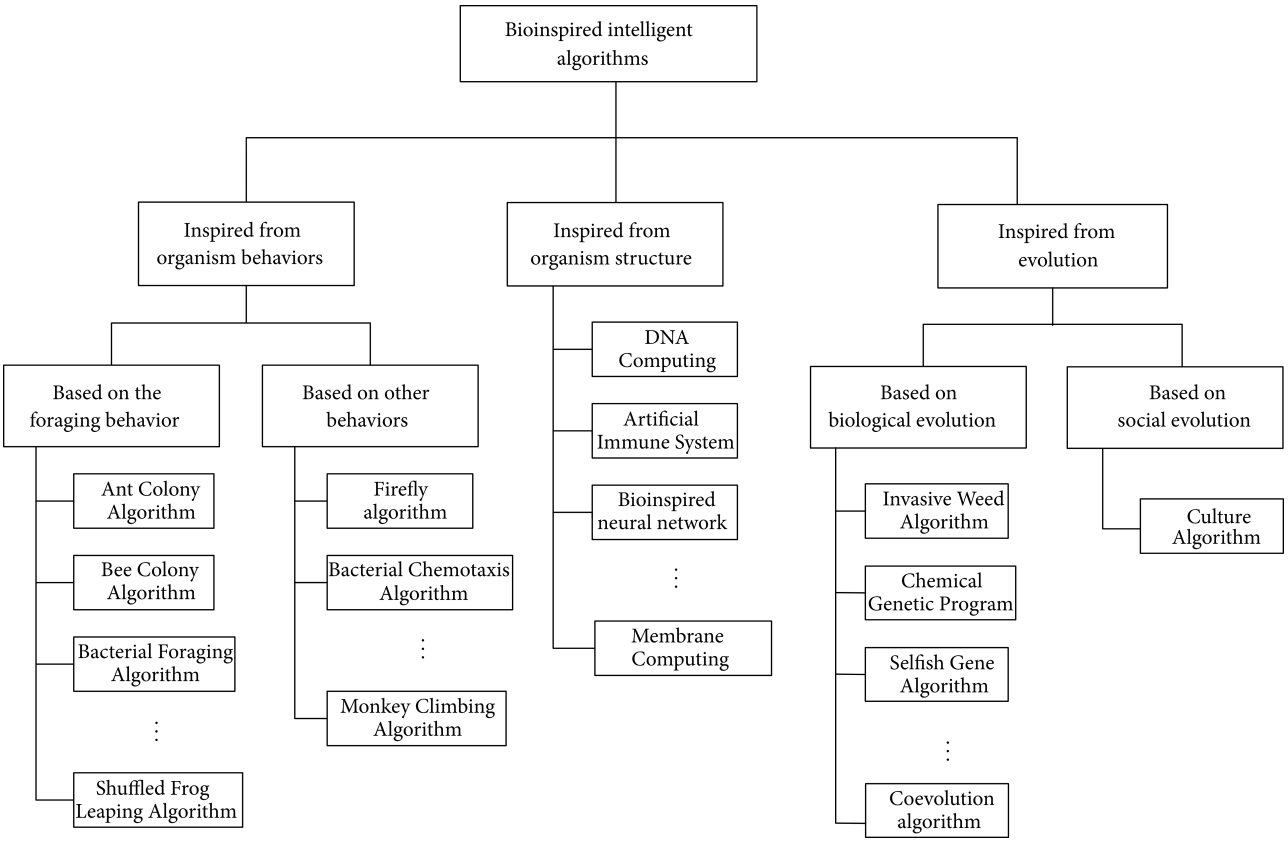

Figure 2.10: Classification of BIAs[9]

2.5.2 Fire hawk optimizer

Inspiration

Native Australians have long used fire to maintain the ecological and topographical balance of their surroundings and have included it into their cultural and ethnic traditions. The vulnerability of the local ecology and biodiversity may increase due to people and other factors spreading lightning- or intentionally set or naturally occurring fires. In addition, it has recently been discovered that brown falcons, whistling kites, and black kites have the ability to start significant fires in the area. The previously mentioned birds, often referred to as Fire Hawks, actively spread fire by carrying flaming sticks in their beaks and talons, a behavior that is considered to be a calamity. In Fig. 2.10 , we are able to observe how these birds react to fires.The birds build small fires to contain and capture their prey by picking up flaming sticks and dropping them in other unburned areas. These tiny flames startle the prey, including snakes, rodents, and other creatures, causing them to flee quickly and in a panic, which makes it much easier for the hawks to catch them[10].

Figure 2.11: Photographs of Fire Hawks' behavior around fires[10]

2.5.3 Strength of Fire Hawk optimizer

The algorithm is a type of metaheuristic algorithm, which is a problem-solving approach that tries to find the best solution by iteratively searching through a large solution space [10].

The Fire Hawk Optimizer algorithm can be summarized in the following steps:

- Inspiration from Nature: FHO draws inspiration from the foraging behavior of whistling kites, black kites, and brown falcons, known as Fire Hawks. By mimicking their natural hunting strategies, FHO capitalizes on the inherent efficiency and effectiveness of these predatory birds. This nature-inspired approach allows FHO to benefit from millions of years of evolutionary fine-tuning, making it well-suited for solving complex optimization problems.

- FHO can outranks the compared algorithms in the selected mathematical test functions, by converging to the predefined tolerance of the global best in a faster and more efficient way.

- Robustness: Bio-inspired algorithms such as FHO often exhibit robustness in dealing with noisy and uncertain data. They can handle complex and dynamic environments effectively, making them suitable for real-world applications where the problem space is unpredictable.

- Fire Hawk Optimizer promote diversity in it's search process, exploring multiple solution candidates simultaneously. This diversity increases the chances of finding novel and innovative solutions, facilitating exploration of the solution space effectively.

- Initialize the algorithm by setting the parameters, such as the population size, number of iterations, and search range.

- Fast Convergence: FHO demonstrates remarkable convergence speed, surpassing the performance of alternative metaheuristic algorithms . It converges to the global best solution in a shorter time frame, enabling it to find optimal or near-optimal solutions more quickly. This efficiency is particularly valuable in time-sensitive applications or scenarios where computational resources are limited.

- Evaluate the fitness of the new population and repeat the process until a stopping criterion is met (e.g., a maximum number of iterations is reached or the improvement in the fitness of the best solution is below a certain threshold).

- FHO is able to converge to the global best of the mathematical test functions by requiring a lower number of objective function evaluations, which proves its efficiency from a computational point-of-view.

- By emulating biological systems and processes, FHO provide a source of inspiration for novel problem-solving approaches. They offer new perspectives on complex problems and inspire the development of innovative algorithms and techniques.

Mathematical Model

The FHO algorithm mimics the hunting habits of fire hawks by taking into account both the process of catching prey and initiating and spreading fires. Based on the location vectors of the fire hawks and their prey, a collection of potential solutions (X) is initially established. The initial placements of these vectors in the search space are determined using a random initialization technique.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^d \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^d \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^j & \cdots & x_N^d \end{bmatrix}, \quad \begin{cases} i = 1,2,\dots,N. \\ j = 1,2,\dots,d. \end{cases} \tag{1}$$

$$x_i^j(0) = x_{i,min}^j + \text{rand.}\left(x_{i,max}^j - x_{i,min}^j\right), \quad \begin{cases} i = 1,2,\dots,N. \\ j = 1,2,\dots,d. \end{cases} \tag{2}$$

$$x_i^j(0) = x_{i,min}^j + \text{rand.}\left(x_{i,max}^j - x_{i,min}^j\right), \begin{cases} i = 1,2,\dots,N. \\ j = 1,2,\dots,d. \end{cases}$$

In Figure 2.12 , the pseudo-code of the FHO algorithm is provided, and The figure 2.13 presents the flowchart of this algorithm[10].

```
procedure Fire Hawk Optimizer (FHO)
        Determine initial positions of solution candidates (X_i) in the search space with N candidates
        Evaluate fitness values for initial solution candidates
        Determine the Global Best (GB) solution as the main fire
        while Iteration < Maximum number of iterations
            Generate n as a random integer number for determining the number of Fire Hawks
            Determine Fire Hawks (FH) and Preys (PR) in the search space
            Calculate the total distance between the Fire Hawks and the preys
            Determine the territory of the Fire Hawks by dispersing the preys
            for l=1: n
                Determine the new position of the Fire Hawks
                for q=1:r
                    Calculate the safe place under lth Fire Hawk territory
                    Determine the new position of the preys
                    Calculate the safe place outside the lth Fire Hawk territory
                    Determine the new position of the preys
                end
            end
            Evaluate fitness values for the newly created Fire Hawks and preys
            Determine the Global Best (GB) solution as the main fire
        end while
        return GB
end procedure
```

Figure 2.12: Pseudo-code of FHO[10]

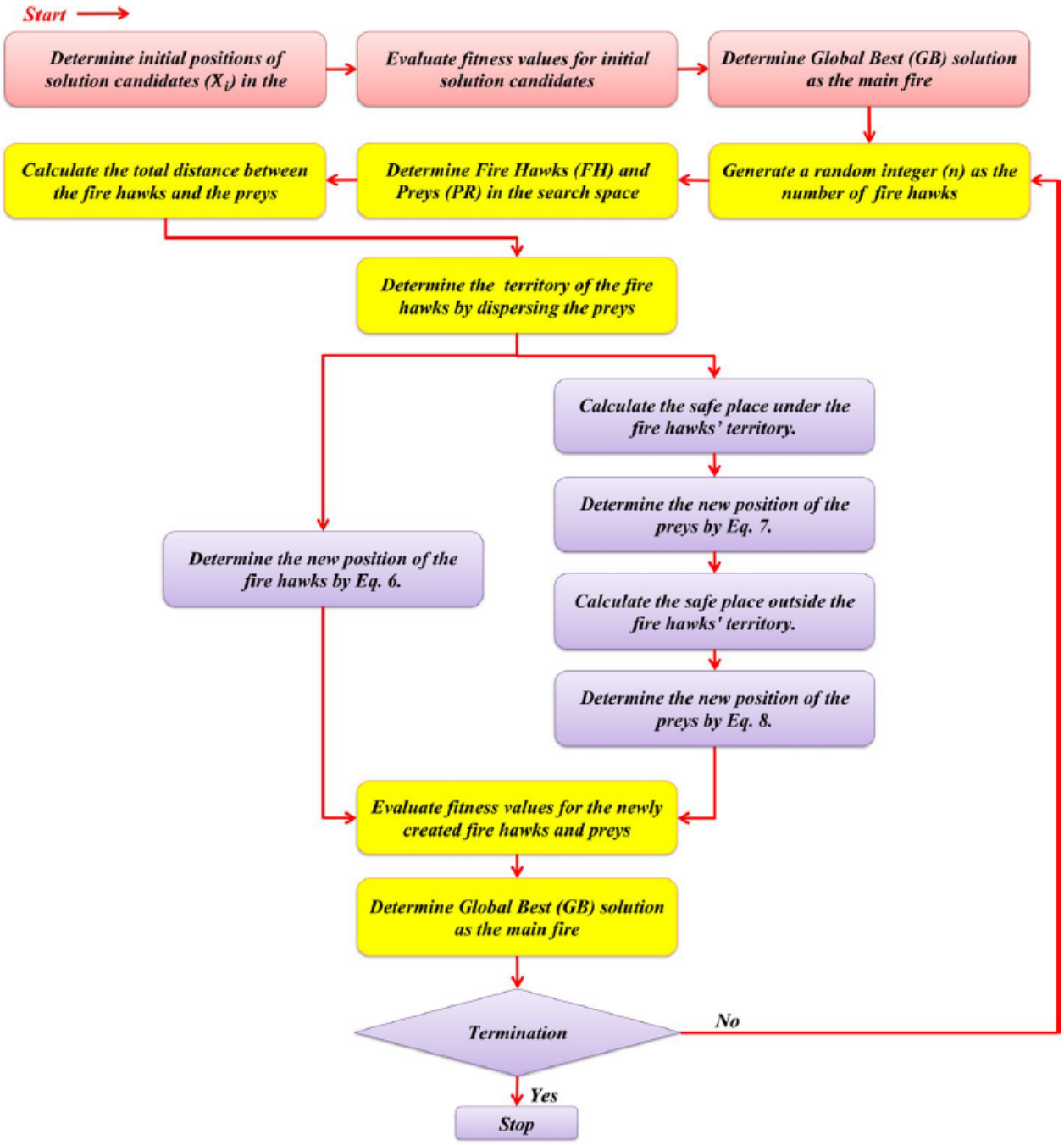

Figure 2.13: Flowchart of FHO[10]

2.6 Conclusion

In conclusion, this section has provided an overview of the key concepts that form the foundation of our work. We discussed online social networks, which are digital platforms facilitating social interactions, as well as false accounts, referring to deceptive or fraudulent user profiles. We also explored machine learning, encompassing algorithms that enable computers to learn and make predictions, and metaheuristics, which are optimization algorithms seeking optimal solutions to complex problems. Additionally, we introduced the Fire Hawk Optimizer, a bio-inspired metaheuristic algorithm designed to tackle optimization problems using unique characteristics and mechanisms. By understanding these concepts, we can delve further into the research and applications at the intersection of online social networks, false accounts, machine learning, and metaheuristics.

Chapter 3

Fake Profiles Detection on Social Networks : State of the art

3.1 Introduction

In this section, we go over a literature study on methods for spotting false profiles.

The number of individuals accessing various social media is rising alarmingly as social networks become the rage of the current era. A significant amount of data has been shared and stolen as a result of this rise.

This chapter reviews the literature on spam review detection using the spammer behavioral features analysis technique. Many different approaches for spotting imposter accounts have been developed by various researchers. Also, by contrasting this new work with earlier research, this study seeks to evaluate its contribution.

Several counterfeit identification techniques concentrate on social network and profile analyses to categorize characteristics or variances that aid in differentiating between genuine and bogus accounts. Algorithms are used to categorize the retrieved data from profiles and posts in particular to create a categorization for the detection of fraudulent accounts.

1. **A novel machine learning-based framework for detecting fake Instagram profiles 2022**

   The article proposes a novel machine learning-based framework for detecting fake Instagram profiles. The framework entails gathering a sizeable dataset of Instagram profiles, extracting significant attributes, training a machine learning model, assessing its performance, and deploying it to identify and delete fraudulent profiles. The authors contend that by correctly recognizing phony profiles, their system can enhance the Instagram user experience. In terms of identifying spam and bogus profiles, the suggested neural network model has precision and accuracy rates of 93% and 91%, respectively. These outcomes show how well the suggested algorithm works to spot fake Instagram profiles.[24] .

	Precision	**Recall**	**F1-score**	**Support**
0	0.89	0.93	0.91	60
1	0.93	0.91	0.88	60
Macro average	0.91	0.91	0.91	120
Weighted average	0.91	0.91	0.91	120

Table 3.1: classification report [24]

2. **An Approach to Detect Fake Profiles in Social Networks Using Cellular Automata-Based PageRank Validation Model Involving Energy Transfer 2022**
   This study suggests a unique method for identifying fraudulent profiles in social networks using a PageRank validation model based on cellular automata that involves energy transfer. By locating and eliminating fraudulent profiles, this strategy aims to enhance the overall user experience on social networks. By taking into account elements like the quantity of connections, degree of activity, and caliber of material, the model simulates the behavior of actual users in social networks. By analyzing their behavior and the effects they produce, fake profiles can be found via the energy transfer process. The suggested method showed excellent accuracy in identifying phony profiles and calculating user influence in social networks. Traditional centrality methods failed to measure user influence as accurately as the energy-based influence score did. [11] .

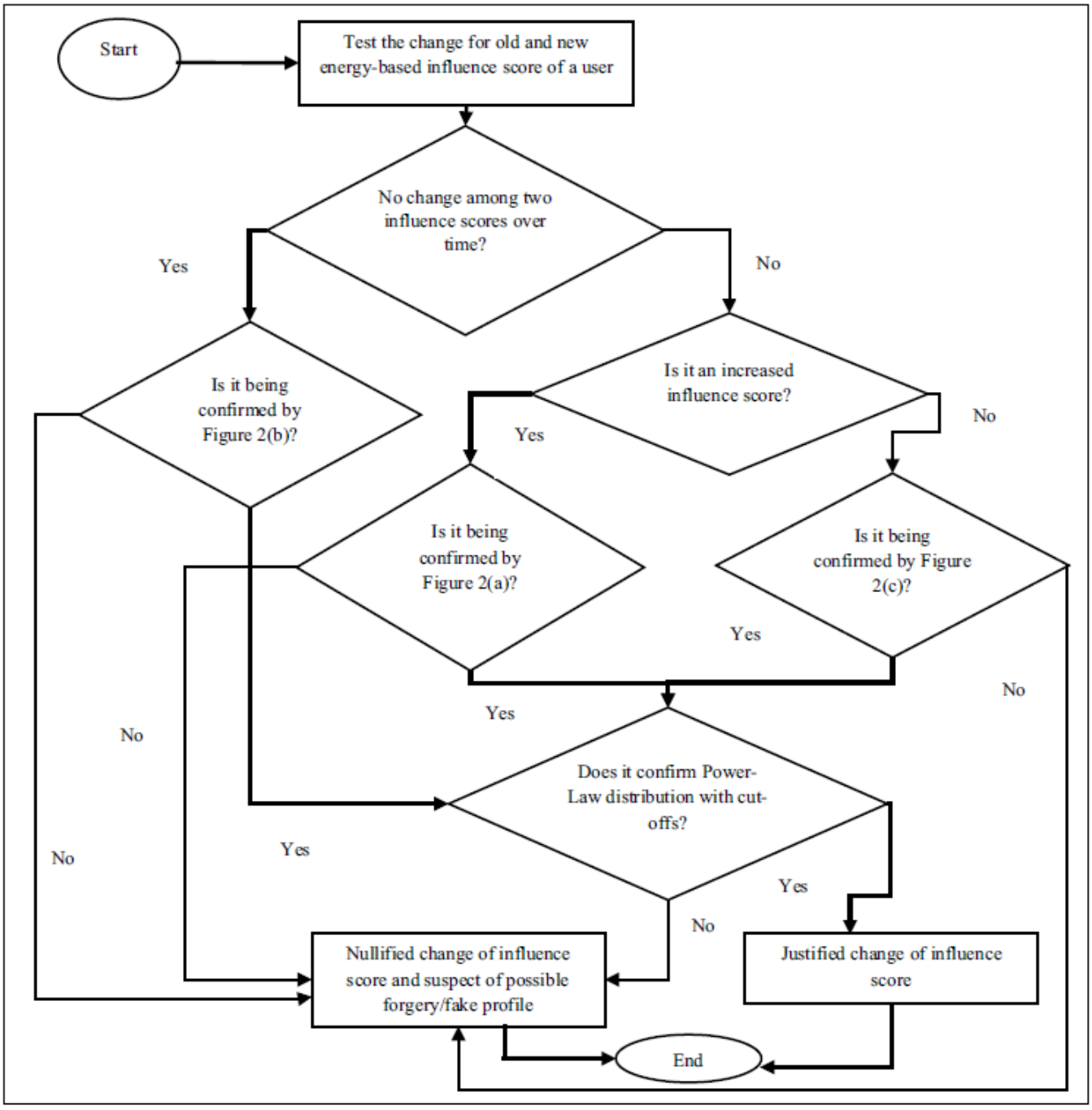

Figure 3.1: Proposed fLowchart of the system [11]

3. **An across online social networks profile building approach: Application to suicidal ideation detection 2022**

   The study provides a unique approach for detecting suicidal intent across diverse social networks by utilizing a thorough profile building approach and NLP tools. The evaluation results show that the suggested approach outperforms existing approaches in detecting suicidal ideation. The approach could have ramifications for mental health researchers and practitioners, allowing for early intervention and prevention of suicidal ideation. [25].

Model	Number of selected features	Accuracy
LR	36	81.4%
SVM with linear kernel	39	79.3%
RF	73	76.2%
XGBoost	28	88.3%

Table 3.2: RFECV results of base classifiers [25]

4. **Automatic Detection of Deaths from Social Networking Sites 2022**
   The development of a natural language processing (NLP)-based system for detecting the deaths of social media users from their online postings and comments is described in this article. To examine language traits and practices in pre- and post-mortem materials, the proposed approach employs machine learning approaches, including both standard and deep learning models. The BERT model beats other methods in detecting fatalities from online posts, according to the study. It also finds substantial variations in the language aspects of pre- and post-mortem tweets, such as the frequency of negative emotions in post-mortem tweets and increased use of personal pronouns, verbs, and death-related phrases. [12].

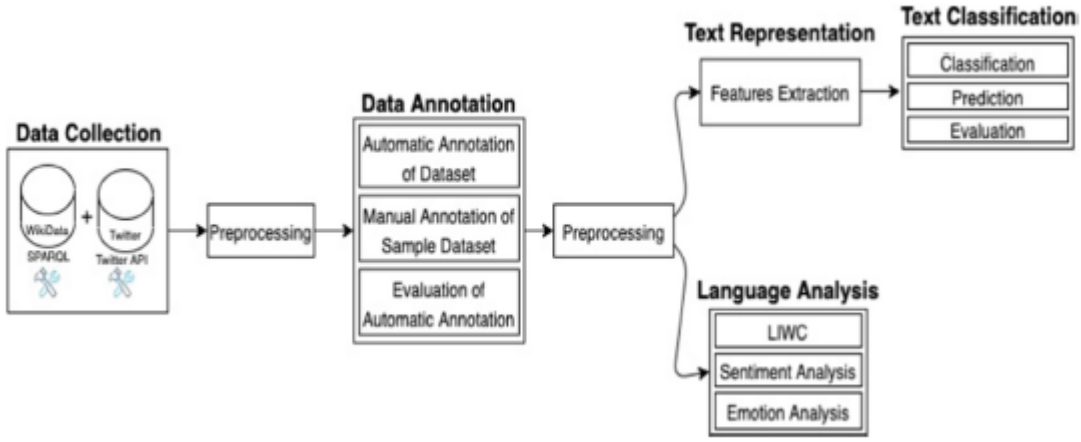

Figure 3.2: Steps involved in our methodology [12]

5. **Deception detection on social media: A source-based perspective 2022**
   Attempting to detect fake information using content-based and context-based approaches. This article focuses on the critical need to prevent the spread of misinformation on online social networks, which poses a serious threat to society.The suggested approach is a source-based strategy that analyzes the profile and interactions of news spreaders on Twitter using a machine learning framework. The research presents four network features and four user-profile parameters for accurately detecting deceit. On two real-world datasets, the XG Boost model beats other models, reaching 92% and 91% accuracy, respectively. The proposed approach takes little information and can detect dishonesty early on. [26].

Model	PolitiFact				GossipCop			
	14 Features		8 Features		14 Features		8 Features	
	Acc%	AUC	Acc%	AUC	Acc%	AUC	Acc%	AUC
Random Forest	82	0.87	89	0.90	83	0.88	87	0.89
Decision Tree	82	0.87	89	0.91	82	0.84	87	0.88
XG Boost	90	0.92	92	0.94	88	0.91	91	0.93
MLP	78	0.84	84	0.86	77	0.82	81	0.84
BernoulliNB	76	0.80	79	0.82	73	0.78	77	0.79
KNN	78	0.76	85	0.79	80	0.78	84	0.78
SVM (Linear)	77	0.73	81	0.77	76	0.75	79	0.77
SVM(RBF)	76	0.76	80	0.79	76	0.75	78	0.79
SVM(Poly)	74	0.70	79	0.74	72	0.73	77	0.76

Table 3.3: Comparing the classification performance of the supervised machine learning models [26].

6. **Detection and Classification of Genuine User Profile Based on Machine Learning Techniques 2022**

   This paper addresses the creation of a machine learning model for identifying and classifying legitimate and fraudulent user accounts on online social networks. The study emphasizes the need of distinguishing authentic user profiles from those of cyber criminals in order to prevent various cyber scams. In all three datasets, the model attained an average accuracy of 95% in categorizing user profiles as authentic or not real. [27].

Classifier	Precision	F1-score	Accuracy	Recall
Random Forest	0.96	0.96	95%	0.95
Neural Network	0.97	0.93	92%	0.89
SVM	0.90	0.901	0.901	0.901

Table 3.4: Comparing performance of classifiers for Twitter dataset [27].

7. **Detection of Fake and Clone Accounts in Twitter Using Classification and Distance Measure Algorithms 2022**

   Because of the rising use of Online Social Networks (OSN), there has been an increase in cyberattacks, such as profile cloning, which allows criminals to steal real users' identities. This problem mostly affects teenagers and women, who are especially vulnerable to cyberattacks. Hackers create replica profiles by stealing the personal information of current users. Fake profiles are also created in cross-site platforms, which differ from the popular OSN walls. Additionally, the lack of authorization information during registration makes it easier for hackers to generate millions of bogus identities on OSNs. against detect profile cloning, this study compares the naive Bayesian probability approach against a KNN algorithm. The results show that the proposed KNN algorithm outperforms the naïve Bayesian probability algorithm in terms of accuracy 70.5 % to 65.0% [28].

	Accuracy	Precision	Recall	F-Measure	Support	Confusion matrix
KNN	70.5	77	56	65	97	[[54 43] [16 87]]
Naïve Bayesian	65.0	66	65	65	200	[[75 22] [48 55]]

Table 3.5: Performance metrics of KNN and NB [28].

8. **Effective Spam Bot Detection Using Glow Worm-Based Generalized Regression Neural Network 2022**

   A. Praveena and S. Smys' paper Effective Spam Bot Detection Using Glow Worm-Based Generalized Regression Neural Network. The suggested approach outperforms the deep Q-learning process in terms of evaluation measures such as accuracy and F-measure, owing to optimal feature training obtained by lowering the classifier's error rate. As a result, the suggested glowworm-based GRNN technique provides an excellent means of determining the nature of Twitter users. [13].

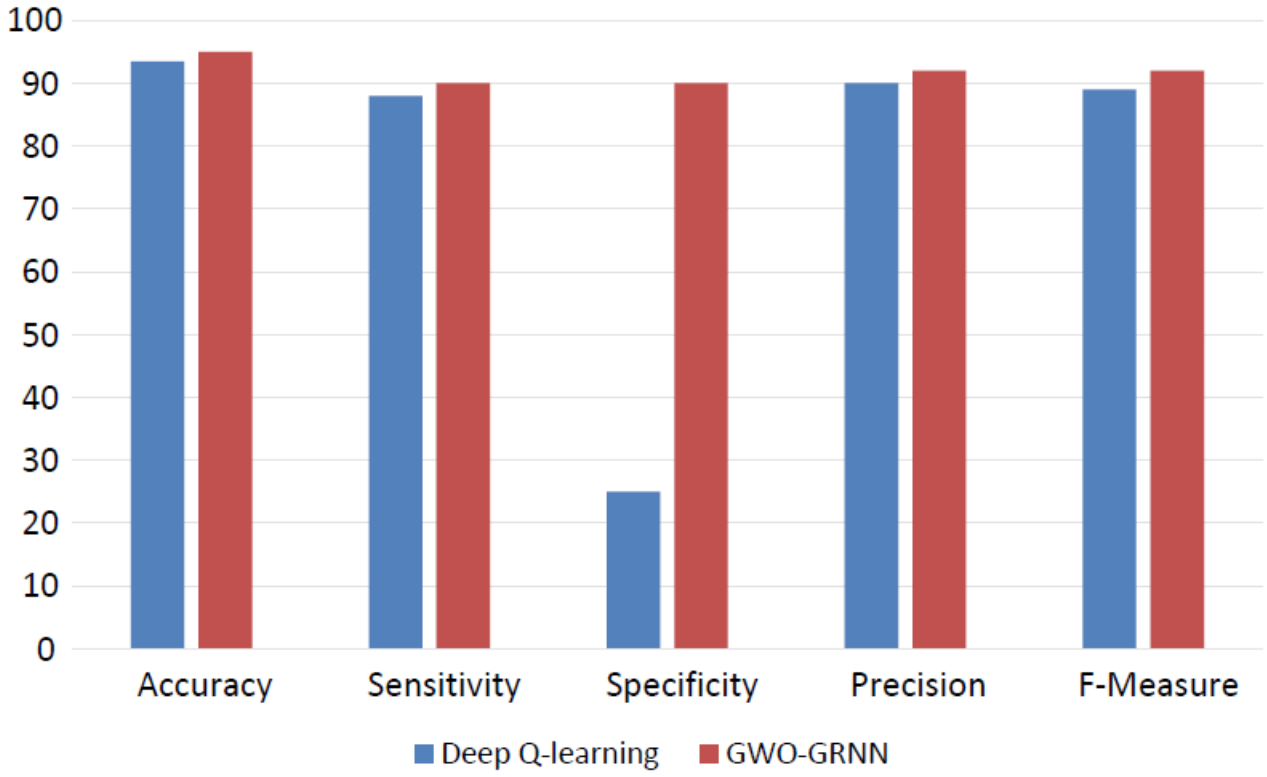

Figure 3.3: Comparison of DQL versus GWO-GRNN for spam bot detection [13]

9. **Fake accounts detection system based on bidirectional gated recurrent unit neural network 2022**

   This study proposes a fake account detection system based on the bidirectional gated recurrent unit (BiGRU) model, which classifies Twitter user profiles as authentic or false based on the content of their tweets. To maintain the semantic and grammatical context, the tweets are translated into a vector space using the global vectors (GloVe) word embedding technique. Long short-term memory (LSTM) and convolutional neural networks (CNN) are baseline models that are outperformed by the suggested strategy., achieving 99.44% accuracy and 99.25 % precision [29].

Classifiers	Accuracy	Precision	Recall	AUC
BiGRU	99.44%	99.25%	99.62%	99.44%
LSTM	98.49%	97.41%	99.62%	98.49%
LSTM+BiGRU	98.68%	99.23%	98.11%	98.68%
CNN	98.87%	98.14%	99.62%	98.87%
CNN+BiGRU	98.87%	99.23%	98.49%	98.86%

Table 3.6: Comparison of model performance [29].

10. **Fake profile recognition using big data analytics in social media platforms 2022**
This article discusses how online social media platforms are becoming a breeding ground for fraudulent personas, which are causing significant harm to both social and economic institutions. While various solutions have been offered to address this issue, the most of them are old and insufficiently accurate, with an average accuracy of only 83%. In answer to this challenge, a Spark ML-based project with more accuracy than previous approaches has been presented. The project is built on Spark ML libraries like Random Forest Classifier and other plotting tools, and it is intended to provide understandable visualizations of the results using diagrams like confusion matrices, learning curves, and ROC plots. The proposed model has been found to have an accuracy of 93% in detecting fake profiles on social media platforms, with a false positive rate of 7% [14].

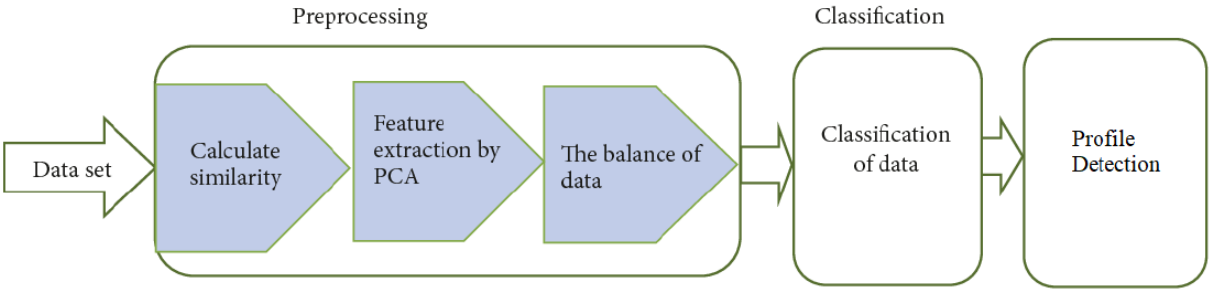

Figure 3.4: Fake profile detection process [14].

11. **Fake Profiles Identification on Social Networks With Bio Inspired Algorithm 2022**
This paper presents a hybrid strategy for detecting false profiles on social media by merging a machine learning algorithm and a bio-inspired algorithm by Nadir Mahammed, Souad Bennabi, Mahmoud Fahsi, Badia Klouche, Nadia Elouali, and Chourouk Bouhadra. The suggested method comprises of two steps and makes use of a dataset from the Facebook social network. The Satin Bowerbird Optimization algorithm is used in the first stage to determine the best bower, which is then utilized as an initial centroid within the k-means clustering approach in the second stage. In terms of detecting accurate profile kinds, the suggested strategy outperforms well-known machine learning algorithms. The hybridization of the two algorithms improves the accuracy of detecting false profiles. [30].

Algorithm	Accuracy [3]	Accuracy
ID3	0.9776	0.9704
SVM	0.9572	0.9564
KNN (k=3)	0.9145	0.9483
NB	-	0.9827
RF	-	0.9765
k-means	0.6731	0.6597
SBO+k-means	-	**0.9892**

Table 3.7: Accuracy comparison [30]

12. **RunMax: fake profile classification using novel nonlinear activation in CNN 2022**
In 2022, Putra Wanda1 published a study offering a strategy for detecting bogus accounts on online social networks (OSNs) using deep learning. The authors emphasize the issue of bogus accounts on OSNs, which can be exploited to disseminate false information and dangerous content. To overcome this issue, they offer RunFake, a dynamic convolutional neural network (CNN) that employs a new activation function called RunMax. When compared to typical functions, this activation function improves training and testing accuracy. When compared to previous approaches, the authors show high precision, recall, and F1-score, as well as a higher area under the curve (AUC). [31].

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

Table 3.8: Comparison of precision, recall and F1-Score among algorithms [31]

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

    Fake accounts that are run by fake identities are causing many problems within the Sudanese online community on Facebook. While research is focused on automatic and semiautomatic accounts, human fake accounts are often neglected. A general characterization of these accounts needs to be considered based on the cultural context they exist within. This research interviewed 250 Sudanese persons on Facebook who fell victim of eight of these accounts. Data was manually harvested for both confirmed fake accounts and confirmed real accounts. The dataset included 231 instances which was imbalanced and skewed toward the real accounts. Over-sampling with SMOTE was applied to treat the over-fitting problem of the machine learning models. Supervised classification algorithms achieved an accuracy of up to 89.7% and an AUC of 0.96 in detecting fake accounts [32].

	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

Table 3.9: Results with and without over-sampling [32]

14. **Online Fake Logo Detection System 2023**

Using machine learning techniques, the study provides a method for recognizing bogus logos. The authors emphasize the growing frequency of online fraud and the use of bogus logos to fool consumers, as well as the necessity for effective tools to detect and prevent the usage of bogus logos on the internet.

The suggested method entails collecting characteristics from logos and building a classifier to differentiate between actual and false logos. The authors tested their method on a dataset of real and fake logos and confirmed its efficacy in recognizing false logos with high accuracy. [15].

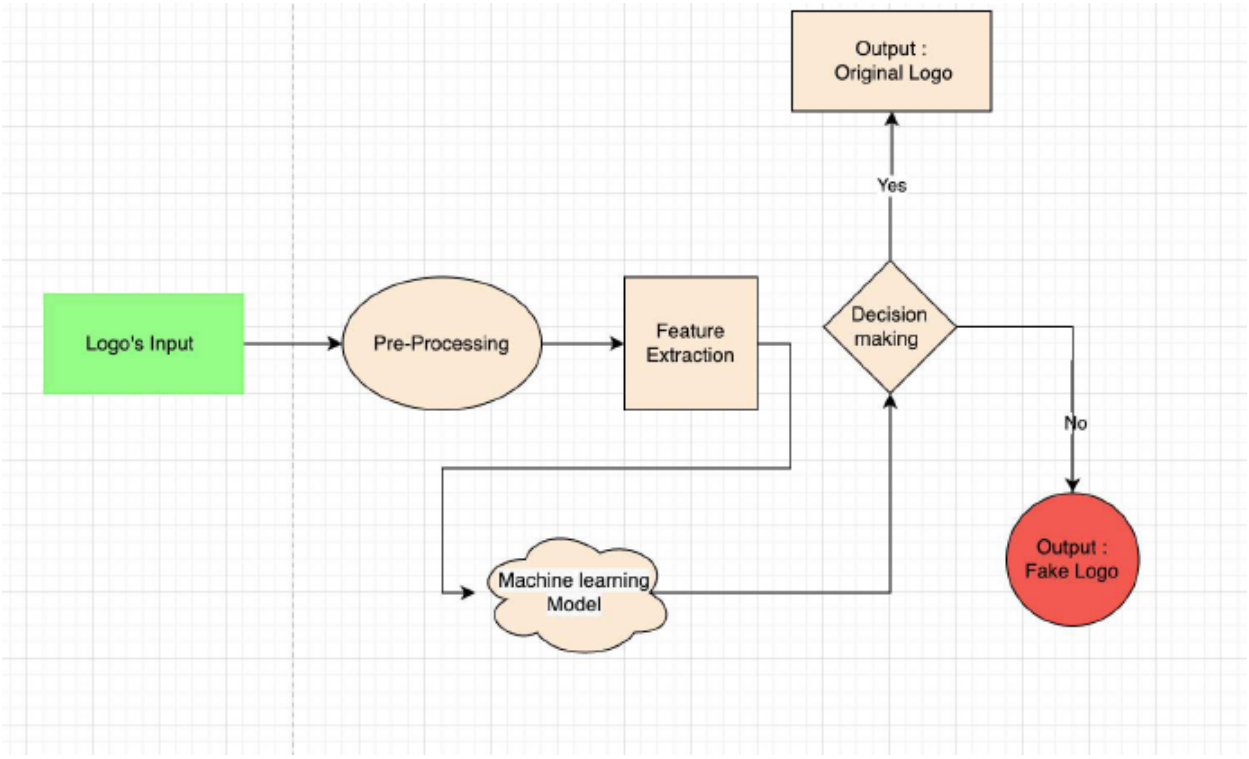

Figure 3.5: Design flow of fake logo detection [15]

15. **Preventing profiling for ethical fake news detection 2023**

    Liesbeth Allein, Marie-Francine Moens, and Domenico Perrotta contributed to this study. The authors used their suggested algorithm to three prominent neural classifiers and found favorable results on false news data discussing diverse news subjects, indicating the feasibility of the proposed goal functions for including social context in text-based classifiers. The application of statistical visualization and dimension reduction approaches revealed that user-inspired classifiers in their latent areas better discern between unseen fake and authentic news. [16].

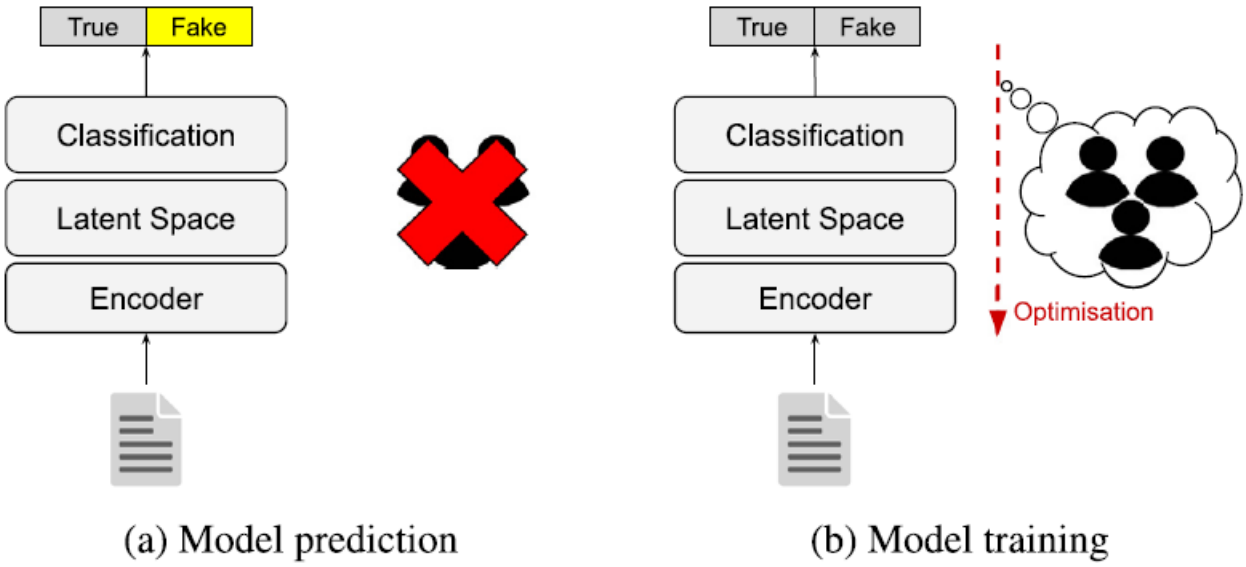

Figure 3.6: The approach taken to detect fake news in an ethical and user-inspired manner[16]

16. **Artificial intelligence applications in fake review detection: Bibliometric analysis and future avenues for research 2023**

A study by (Sami Ben Jabeur, Hossein Ballouk, Wissal Ben Arfi, and Jean-Michel Sahut) gives a bibliometric investigation of the usage of AI in detecting bogus reviews. The research identifies the major advancements, hotspots, and trend directions in this sector. According to the findings, there is a growing research interest in AI solutions for detecting false reviews.This study sheds light on the significance of artificial intelligence in detecting false reviews and highlights critical topics for future investigation. Decision-makers can improve the regulation and control of phony reviews by understanding the current state of research, thereby enhancing the dependability and trustworthiness of online reviews. [17].

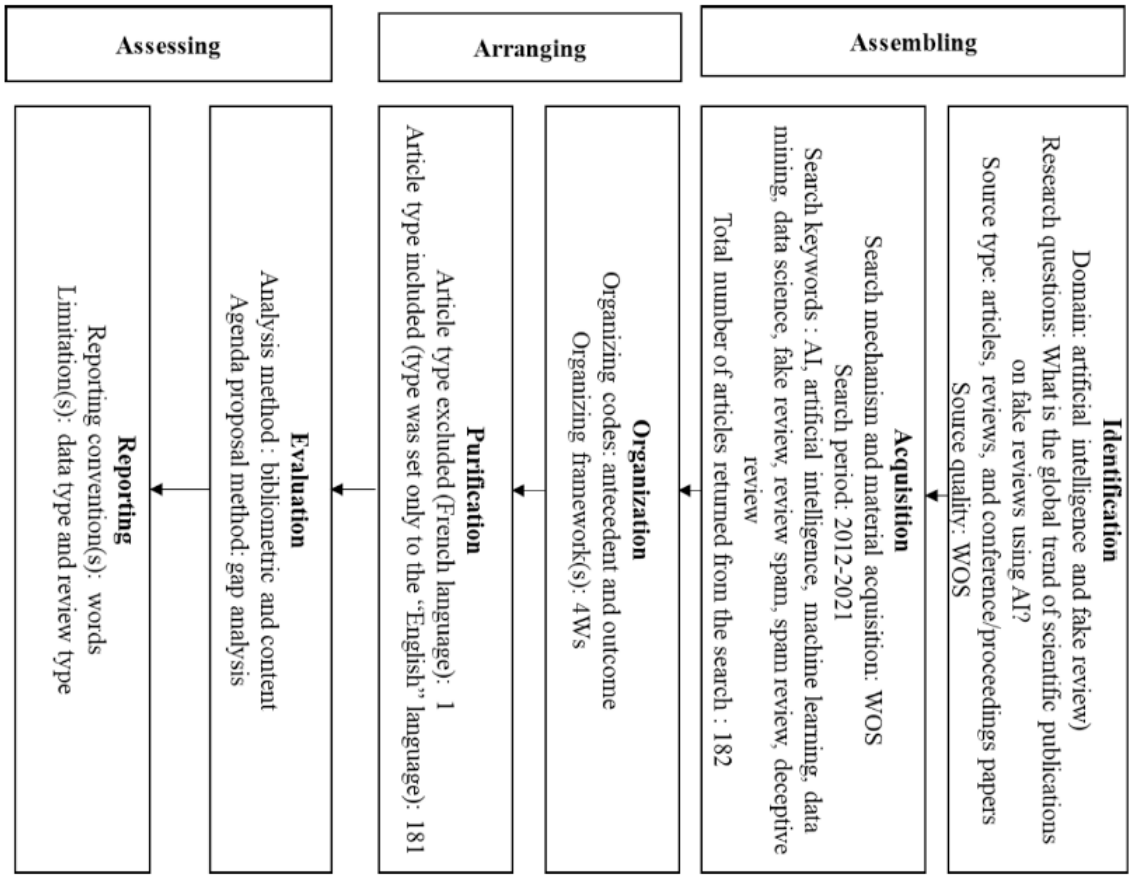

Figure 3.7: Application of the SPAR-4-SLR protocol[17]

17. **Deep Learning Hybrid Approaches to Detect Fake Reviews and Ratings 2023**
This paper presents two unique deep-learning Hybrid techniques: CNN-LSTM for detecting phony online reviews and LSTM-RNN for detecting fake ratings in e-commerce. To detect false reviews, the CNN-LSTM technique use a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. The CNNs collect the most relevant features from the review text, while the LSTM network models the temporal connections between the extracted features. For word embedding, the approach also employs Glove and One-hot encoding techniques. To detect false ratings, the LSTM-RNN technique employs a combination of LSTM networks and Recurrent Neural Networks (RNNs). The LSTM network models the temporal connections between rating scores, whereas the RNN detects patterns in rating sequences. Both hybrid models beat existing methods in detecting false online reviews and ratings, according to the experimental data. The CNN-LSTM model has the highest prediction accuracy in detecting phony reviews, while the LSTM-RNN model has the highest precision in detecting fraudulent ratings, with a precision of 93.8 percent. The experimental investigation successfully demonstrated that the CNN-LSTM and LSTM-RNN may be highly effective and practical in detecting false online reviews and ratings. .[33].

Classification Word embedding model	Accuracy	Precision	Recall	F1-score
Linear SVM + TF IDF	92%	92%	90%	90%
Decision tree + TF IDF	92.80%	92.30%	92.30%	92.80%
1st Proposed Model CNN-LSTM +GloVe	93.07%	93.09%	93.07%	93.07%
2nd Proposed Model LSTM-RNN + One Hot Encoding	93.09%	93.09%	93.08%	93.08%

Table 3.10: Accuracy of a new hybrid model [33]

18. **Deepfakes: Deceptions, mitigations, and opportunities 2023**

The following piece addresses the rise of deepfakes, which are digitally modified audio, video, or images that can confuse consumers and harm businesses. The essay investigates the threats that deepfakes bring to organizations and consumers, as well as potential applications for the technology. To combat these concerns, businesses should invest in detection and prevention tools, while consumers can be wary of information sources. Despite the risks represented by deepfake technology, this new technology also offers benefits. Future research directions suggested in the study include investigating the legal and ethical aspects of deepfakes and developing new detection tools. [18].

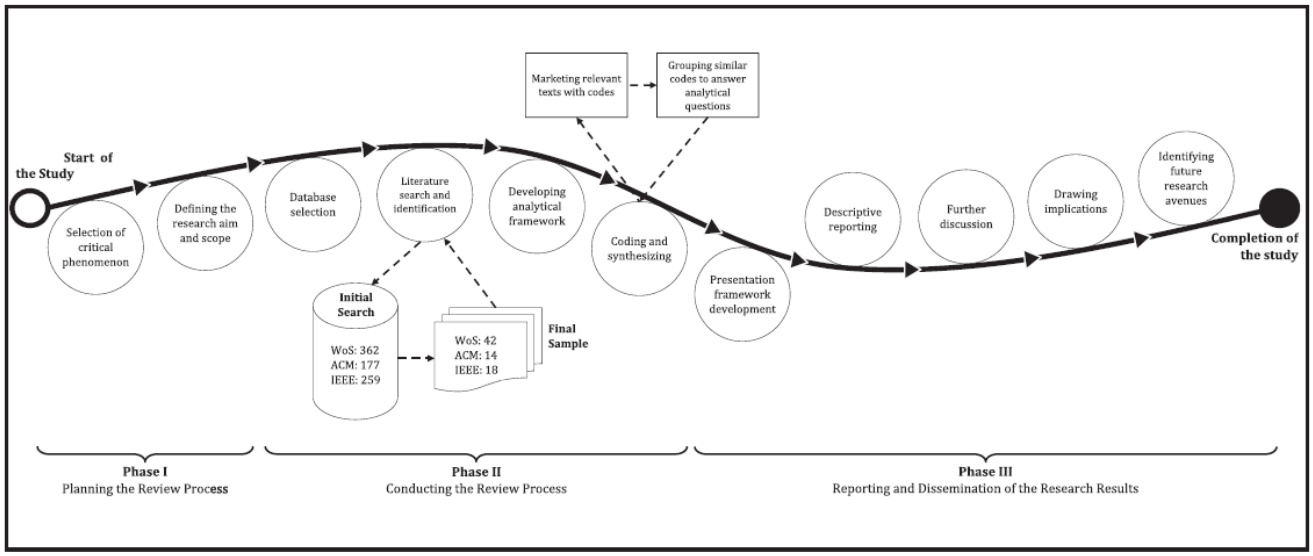

Figure 3.8: Visual illustration of the overall process of this study [18].

19. **Distance-based customer detection in fake follower markets 2023**

Boyeon Jang, Sihyun Jeong, and Chong-kwon Kim's article (Distance-based consumer detection in false follower marketplaces) investigates the relationship between geographic distance and online social ties, specifically on Twitter. The authors compiled a dataset from Twitter that included ground truth for 7933 authentic users, 24,552 fraudulent followers, and 2341 consumers. They discovered that when the distance between two nodes rises, the association likelihood of legitimate users drops, which is consistent with the small world phenomena. Customer links, on the other hand, do not follow this pattern. They also discovered that when distance grows, the percentage of followers for consumers increases, while the percentage of followers for real users falls. On the basis of this discovery, the authors proposed a method for detecting fraudulent follower consumers by adjusting geographic distance. They used follower distance ratio to rank customers and compared their technique to existing representative algorithms for false followers and customer detection. . Their approach showed better performance and a high percentage of accuracy (98.1

	Accuracy	FPR	FNR	F1-score	MCC
Our Method	0.981	0.017	0.027	0.958	0.946
CatchSync	0.935	0.037	0.157	0.856	0.814
DetectVC	0.952	0.031	0.104	0.895	0.865

Table 3.11: Comparison of Experiment results in the entire dataset [34]

20. **Detection and Verification of Cloned Profiles in Online Social Networks Using MapReduce Based Clustering and Classification 2023**

This research by (Saravanan A, Vineetha Venugopal) explores the problem of cloned and false profiles on online social networks and provides a new method for detecting such profiles with greater precision and accuracy. Clustering and classification techniques, as well as other relationship features, are used in the suggested strategy. The authors built a dataset to evaluate the performance of the suggested strategy, utilizing multiple forms such as the Facebook Graph API, manual search, and extraction of mutual friends' details. The dataset used in the study includes 1,000 records, 400 of which were real samples and 600 of which were cloned or fake profiles that were manually classified. The authors compared the performance of their proposed method with other methods and found that it outperformed it with accuracy and accuracy of 98.19% and 98.96% for the MIB twitter data set and 98.90% and 99.17% for the synthetic data set generated for the study. [19].

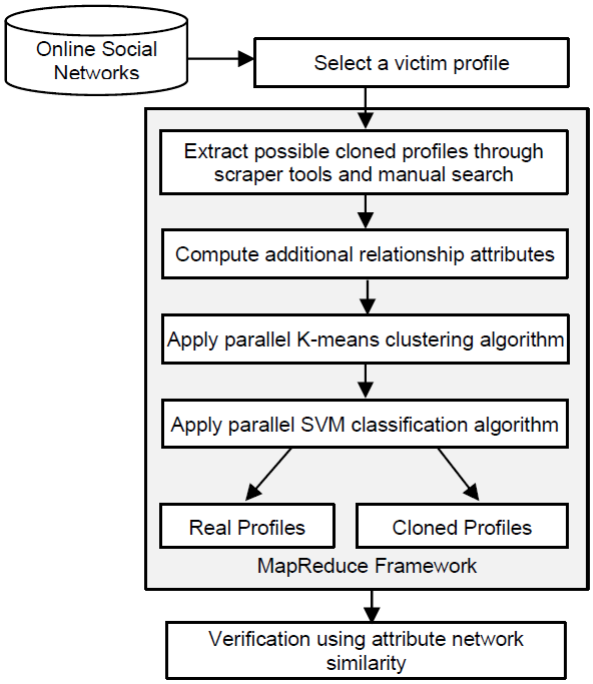

Figure 3.9: Workflow of the Proposed Cloned Profile Detection Model [19]

3.2 Synthesize and discussion

We have summarized the literature presented above in tabular format Table and given a rebuttal of what people have researched in their works between 2022 and 2023 and also we have stated the findings and limitations of the reviewed papers.

Reference	**OSN**	**ML Type**	**ML Algorithms**	**Meta-heuristic**	**Dataset-(Instances)**	**Results**
1	Instagram	Supervised	InstaFake InstaResearch		120 , 576	Acc=95%
2	Youtube	Supervised	Naïve bayes			
3	Twitter Youtube Tumblr	Supervised	RF,NB ,SVM ,DT ,adaB, XGBoost		111	Acc(XG Boost)=88.3% Acc(SVM)=91.5%
4	Twitter	Supervised	RF, KNN, LR, SVM, BiLSTM and CNN			Acc(BERT)=91.60%
5	Twitter	Supervised	XG Boost RF,DT, MLP, SVM,KNN			Acc=95%
6	Facebook Instagram Twitter	Supervised	RF , SVM ,KNN		608 , 2820 , 11420	Acc(RF)=96%.

7	Twitter	Supervised	KNN ,NB		9	Acc(KNN)=90.2% Acc(Nb)=65.5%
8	Twitter	Bio-inspired		GWO , GRNN	19276 ,22223	Acc=95%,
9	Twitter	Unsupervised	BiGRU Glove		2818	Acc(BiGRU Glove)=99.44%
10	Facebook, Twitter	Supervised	SVM, MLPC ,KNN ,RF ,NB		51, 28001, 6194	Acc(RF)=94%
11	Facebook	Unsupervised		Kmeans +SBO	1244	Acc(SBO+K means)=98.9%
12	Facebook Twitter Instagram	Supervised	CNN		20000	Precision=94%, Recall=93.21%, F1-score=93.42%
13	Facebook	Supervised	LR, SVM, DT, NB, RF		231	DT=0.89, RFM=0.89, SVM=0.87, NB=0.85, LR=0.86
14	Instagram	Supervised	CDS (context depen-dent similar-ity)		20000	Acc(min)=90%

15	Twitter	Supervised	CNN, HAN, Distil-BERT		2029	F1score(CNN)=82%, F1score(HAN)=82%, F1score(DB)=87%
16		Supervised	RCNN, RNN			
17		Supervised	CNN-LSTM, LSTM-RNN		20000	Acc(CNN-LSTM)=93.07% Acc(LSTM-RNN)=93.09%
18	Facebook				12000, 3300	
19	Twitter	Supervised	RF		7933, 24553, 2341	Acc=98%
20	Facebook	Supervised	Kmeans, SVM		1000	Acc=98.90%

Table 3.12: Summarized of literature

The table 3.12 presents a collection of references, online social networks (OSNs), machine learning (ML) types, ML algorithms, datasets, metaheuristics, and corresponding results. It showcases a diverse range of studies and approaches employed in the field.

The table demonstrates the variety of OSNs used for analysis, including popular platforms such as Instagram, YouTube, Twitter, Facebook, and Tumblr. These OSNs serve as rich sources of data for ML applications. The ML types employed encompass both supervised and unsupervised learning, allowing for the exploration of different research directions.

Various ML algorithms are employed across the studies, including well-known methods like Naïve Bayes, random forest (RF), support vector machines (SVM), decision trees (DT), and XGBoost. This highlights the researchers' efforts to leverage a wide array of

techniques to tackle different challenges and achieve high accuracy levels, often surpassing the 90

Additionally, the use of bio-inspired algorithms and metaheuristics, such as Grey Wolf Optimization (GWO) and Genetic Recurrent Neural Networks (GRNN), reveals the adoption of nature-inspired approaches to problem-solving. These methods offer promising avenues for optimizing ML models and improving overall performance.

The datasets used vary in size, ranging from small-scale instances to larger datasets with thousands of instances. This demonstrates the researchers' adaptability in handling different data volumes and their dedication to comprehensive analysis.

The presented results showcase impressive achievements, including high accuracy rates, precision, recall, and F1-scores. These outcomes underscore the effectiveness of the applied ML algorithms and techniques in addressing challenges specific to each OSN and ML task.

In summary, this table provides a comprehensive overview of the studies conducted in the realm of OSN analysis using ML approaches. It highlights the researchers' commitment to exploring various ML algorithms, leveraging bio-inspired techniques, and utilizing diverse datasets to deliver impressive results and advance the field of social network analysis.

3.3 Conclusion

We have given a theoretical foundation for understanding OSNs technology in this chapter.

This backdrop enables us to discuss the state-of-the-art research that is being done to address this security issue, as well as the problem of fake profiles.

The following are some key takeaways from this analysis:

- Twitter has become the most targeted OSN platform. Since it's information is made available to the public by default and is simple to obtain using Twitter APIs, Twitter has been the most targeted platform.

- The studies reviewed might use largest dataset to test the fake profile detection on for having enough data which has more data than the algorithm may recognize the profile also recognize the characteristics of fake profile.

- Both supervised and unsupervised machine learning are used to detect bogus profiles. These modules are effective at identifying phony accounts.

- We discovered According to our research, both legitimate and fraudulent accounts typically have a profile picture and name attribute.

Chapter 4

Our contribution

4.1 Introduction

The initial phase of this study involved conducting an in-depth analysis of diverse machine learning algorithms using a comprehensive collection of readily accessible data from Facebook, Twitter, and Instagram profiles. The primary objective was to evaluate the performance of these algorithms based on a range of performance metrics, thus providing valuable insights.

The study's scope was carefully defined, taking into account its objectives and the available resources. The aim was to gain a comprehensive understanding of the behavior of machine learning algorithms when applied to specific datasets extracted from Facebook, Twitter, and Instagram accounts, with the ultimate goal of drawing meaningful conclusions. Subsequently, thorough analysis and comparison of the obtained results with relevant literature allowed for the derivation of insightful findings..

4.2 Data processing

4.2.1 Dataset Collection :

Facebook dataset Description :

We cautiously assembled a comprehensive dataset of Facebook accounts for the planned study. The data was collected by exploiting the capabilities and functions offered by the prominent social networking site Facebook. The final dataset is amazing, with 1244 rows and 15 columns containing a wealth of useful information. To ensure the reliability and relevance of our findings, we employed the meticulously curated Facebook Dataset 9-9-2019. This specific dataset was specifically selected for its comprehensive coverage and up-to-date representation of Facebook accounts. The 1244 accounts in this dataset were carefully categorized and separated 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%

Figure 4.1: Facebook dataset description[20]

From the meticulously compiled Facebook dataset, we extract a diverse array of informative features that contribute significantly to the effective classification of the data at hand. These carefully selected 14 features offer valuable insights and enable us to gain a comprehensive understanding of the underlying patterns and characteristics within the dataset. The chosen 14 features set encompasses a wide range of essential aspects are as seen below

- 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 : Analysis of several Facebook criteria in our system are described below to identify real accounts from fake ones on Facebook :

Features	Description	Justification
Profile Picture	Visual identification of the user.	Real users use their real pictures more often than fake users.
Work place	Workplace or job title's information,	Real users more often use their real workplace information than fake users.
Education	Attended (school, college, university…etc.) information.	Real users mentioned their education information in their Facebook profiles more often than fake users.
Living Place	Living place address (city, town, state…etc.) information.	Real users more often use their real living place information than fake users
Check-In	Information for announcing user location.	Real users check into places in their Facebook's profiles more often than fake users.
No. of Posts	Social online activities shared on Facebook	Real users have more online activities than fake users.
No. of Tags	Identify the user by someone else on his/ her wall.	Real Users tagged more often than fake users.
Introduction "Bio."	Introduction information about Facebook's users.	Real users are more often write something about themselves than fake users.
No. of Mutual Friends	Number of the people who are Facebook friends with both users and the target profiles.	Real users have more mutual friends with target profile than fake users, hence gives profile more incredibility.
No. of Pages	Number of pages liked.	Real users usually liked more pages than fake users.
No. of Groups	Number of groups joined.	Real users usually join groups more than fake users.
Family\ Relationship	Social relation Information\Status	Real users share their real social relation status than fake users.

Figure 4.2: Attributes descriptions and intuative justification[20]

Twitter Dataset

In our previous research, we utilized a dataset generated through the utilization of the Twitter API. The Twitter API encompasses four main objects, namely Tweets, Users, Entities, and Locations. Each of these objects exhibits a diverse array of characteristics. The tweeting objects serve as the essential atomic building pieces at their heart. They have many aspects that correspond to general tweet information, such as the creation time, the number of likes received, the number of retweets, and more. It should be noted, however, that protected accounts cannot access these features. On the other hand, users objects encompass a wide range of entities, representing individuals or entities of any nature. These objects contain attributes that provide insights into the general account information, such as the number of tweets liked, the number of followers, and other contextual metadata.

Entities objects play a crucial role in providing additional contextual information about the content shared on Twitter. They encompass values present within the tweets, including hashtags, media elements, and URLs.

Lastly, locations and items are identified by names that correlate to their corresponding geographic coordinates. The dataset we utilized consists of 16 attributes, further enriching our analysis and understanding of the Twitter platform.

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

Figure 4.3: Twitter dataset description[21]

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.

Figure 4.4: Features of Twiter dataset[21]

Instagram Dataset

The dataset used in this analysis was obtained from Kaggle.com, a popular platform for data science and machine learning enthusiasts. The dataset consists of two CSV files: test.csv and train.csv. These files contain information about user accounts and their authenticity.

The main objective of this dataset is to determine whether a given user account is fake or real. To accomplish this, the dataset includes a dependent variable, which is categorical and has two possible values: 0 for real accounts and 1 for fake accounts. This variable serves as the target or label for our machine learning models.

One important characteristic of the training dataset is its balanced distribution. The data is structured in such a way that 50% of the instances are fake accounts, while the remaining 50% are real accounts. This balanced distribution ensures that our models are trained on an equal number of examples from each class, allowing them to learn effectively

without being biased towards either category.

By using this dataset, we can employ various machine learning algorithms to build models that can accurately classify user accounts as real or fake. These models can then be utilized to detect and prevent the creation of false profiles, which is crucial for maintaining the integrity and security of online platforms..[22]

	A	B	C	D	E	F	G	H	I	J	K	L
1	profile pic	nums/len	fullname	nums/len	name==us	descriptio	external U	private	#posts	#follower	#follows	fake
2	1	0.27	0	0	0	53	0	0	32	1000	955	0
3	1	0	2	0	0	44	0	0	286	2740	533	0
4	1	0.1	2	0	0	0	0	1	13	159	98	0
5	1	0	1	0	0	82	0	0	679	414	651	0
6	1	0	2	0	0	0	0	1	6	151	126	0
7	1	0	4	0	0	81	1	0	344	669987	150	0
8	1	0	2	0	0	50	0	0	16	122	177	0
9	1	0	2	0	0	0	0	0	33	1078	76	0
10	1	0	0	0	0	71	0	0	72	1824	2713	0
11	1	0	2	0	0	40	1	0	213	12945	813	0
12	1	0	2	0	0	54	0	0	648	9884	1173	0
13	1	0	2	0	0	54	1	0	76	1188	365	0
14	1	0	2	0	0	0	1	0	298	945	583	0
15	1	0	2	0	0	103	1	0	117	12033	248	0
16	1	0	2	0	0	98	1	0	487	1962	2701	0
17	1	0	3	0	0	46	0	0	254	50374	900	0
18	1	0	3	0	0	0	0	0	59	7007	289	0
19	1	0.29	3	0	0	48	0	0	1570	1128	694	0
20	1	0	2	0	0	63	1	0	378	34670	1878	0
21	1	0	2	0	0	106	1	0	526	2338	776	0
22	1	0	2	0	0	40	0	0	228	3516	999	0
23	1	0	1	0	0	35	1	1	35	1809	416	0
24	1	0	2	0	0	30	0	0	281	427	470	0
25	1	0	1	0	0	27	0	0	285	759	956	0

Figure 4.5: Snapshot of Training Dataset[22]

Exploratory Data Analysis: This is a vital step of early data exploration that is carried out in order to detect patterns in the dataset and identify abnormalities using summary statistics and graphical representation. The many separate processes completed are shown here.[22]

4.2.2 Missing Value Treatment

There were no missing values in the given dataset. Missing values in a dataset can occur owing to a variety of real-world issues and can be handled by deletion or imputation[22].

The inclusion of missing values limits the amount of data accessible for analysis, reducing the study's statistical power and, ultimately, the dependability of its results.

4.2.3 Outlier Detection

Outliers are extraordinary values that depart from the dataset's average data values. If there are outliers in the dataset, the accuracy suffers dramatically since the training dataset learns from the noise in the data and may produce an over-fit model. We find through rigorous network analysis that the following features exhibited outliers: nums/length username, complete name words, description length, #posts, #followers, and #follows. We used median imputation to deal with these outliers by computing the median of this collection of values[22]. The outliers are not included in the median computation. After calculating the median, we replace all outliers with the estimated median value.

4.2.4 Bivariate Analysis

This is done to gain a better understanding of the relationship and strength of link between two variables. We examined the correlation matrix and concluded that there was no significant multicollinearity between the variables. When creating a logistic regression model, this is one of the assumptions that must be made. We can safely proceed to the algorithms after the data has been pre-processed [22]. We now have a labeled training dataset and can apply supervised learning techniques to convert the input to the output.We examined two prominent categorization methods in this article: Logistic Regression and the Random Forest technique. Each one has been thoroughly described here..

Isis Twitter Dataset

Since the November 2015 Paris Attacks, the developers harvested over 17,000 tweets from 100+ pro-ISIS fanboys around the world. In order to establish powerful counter-messaging strategies against violent extremists at home and abroad, they are collaborating with content creators and influencers.[58]

The dataset includes the following:

- Name
- Username

- Description
- Location
- Number of followers at the time the tweet was downloaded
- Number of statuses by the user when the tweet was downloaded
- Date and timestamp of the tweet
- The tweet itself

Based on this data, here are some useful ways of deriving insights and analysis:

- Social Network Cluster Analysis: Who are the major players in the pro-ISIS twitter network? Ideally, we would like this visualized via a cluster network with the biggest influencers scaled larger than smaller influencers.
- Keyword Analysis: Which keywords derived from the name, username, description, location, and tweets were the most commonly used by ISIS fanboys? Examples include: ”baqiyah”, ”dabiq”, ”wilayat”, ”amaq”
- Data Categorization of Links: Which websites are pro-ISIS fanboys linking to? Categories include: Mainstream Media, Altermedia, Jihadist Websites, Image Upload, Video Upload,
- Sentiment Analysis: Which clergy do pro-ISIS fanboys quote the most and which ones do they hate the most? Search the tweets for names of prominent clergy and classify the tweet as positive, negative, or neutral and if negative, include the reasons why. Examples of clergy they like the most: ”Anwar Awlaki”, ”Ahmad Jibril”, ”Ibn Taymiyyah”, ”Abdul Wahhab”. Examples of clergy that they hate the most: ”Hamza Yusuf”, ”Suhaib Webb”, ”Yaser Qadhi”, ”Nouman Ali Khan”, ”Yaqoubi”.
- Timeline View: Visualize all the tweets over a timeline and identify peak moments

4.2.5 Dataset preprocessing

A lot of information has been gathered from numerous sources, including the internet, questionnaires, and tests, among others. Nevertheless, the majority of the time, the data that must be used are distorted, noisy, and full of missing values.Data preprocessing is

a fundamental step to data analysis and machine learning ,is a set of procedures which transforms raw data into understandable format .

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 considered a few classification algorithms before moving on to the dataset. We also give numerical feature types some thought. The numerical aspects of other categorical features had also been altered.

We attempted to evaluate the most important attributes because the dataset has numerous attributes. We substituted each example's Name-Id in the Facebook dataset with a numerical id in order to exclude attributes like Link and Name-Id that are not statistically significant from our model. Applying various ML algorithms to the dataset is crucial.

To accurately categorize the dataset, filtering is applied. Whether the dataset is accurately classified by the classification method and contains no incorrect or null values.

After that, as part of the data preprocessing stage, we had normalized the data. In order to preserve the information, it is important to convert the dispersed huge numerical values to a common scale of [0,1] without distorting the value range-differences. Several algorithms require this phase in order to properly model the data.

When the features in a dataset have a variety of measuring units, normalization of the data is necessary. Use of normalization, which involves rescaling the chosen characteristics from their original values to the scale of 0 to 1, is a helpful strategy when the data distribution is confusing, unpredictable, or not Gaussian.

Text Classification Model:

The detection process of the textual input data consists of three main parts as illustrated in Figure 4.6.

Natural Language Processing:

used for pre processing the textual data in order for us to have a ready structured data that is easy to understand and process. The process of analyzing the textual data is performed in four main steps: tagging, annotating, co-reference and sentiment analysis.

Word embedding :

We have opted for the N-gram language model that estimates the probability of the last word given the previous words. This choice was motivated by the fact that it showed much better results than the TF-IDF model in.

Classification:

After word embedding, the textual content is now in a numerical form, understandable by the machine, and ready to be used by any classifier as an input.

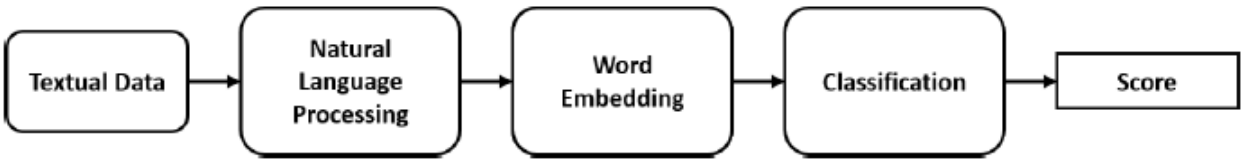

Figure 4.6: Text classification model design[23]

Textual Data

Both detection and prediction models rely on textual data, that is why it is very important to pick the richest datasets.

Positive Labels: We stumbled upon a good amount of datasets that do not fit our needs. We were interested in datasets previously shared among pro-ISIS users in SNs. We finally find out the "How ISIS uses Twitter"3 that contains 17350 tweets from more than 110 pro-ISIS accounts. It includes the following attributes: Name, Username, Description, Location, Number of followers at the time the tweet was downloaded, Number of statuses by the user when the tweet was downloaded, Date and timestamp of the tweet and the tweet itself. Some of the tweets in the dataset are typed in Arabic, thus we used a public translation tool, Google Translate API, each time we detect Arabic characters.

Negatives Labels: We used the Global Terrorism Database (GTD) [10] as part of our negative labeled dataset. The dataset contains information about more than 180000 Terrorist Attacks from all around the word since 1970. We filtered the dataset on events that took place since 2002 and extracted the content of the "summary" column which contains summaries of each of the attacks

$$\begin{cases} S_u > \gamma \Rightarrow Positive(Pro - ISIS) \\ S_u < \gamma \Rightarrow Negative(Anti - ISIS) \end{cases}$$

4.2.6 Features selection

We use feature selection when a dataset contains redundant features or when the outcomes are relatively unaffected by the characteristics.

Correlation matrix

In this section we have used the **correlation matrix** to detect the highest correlation between the features and the class . We use the correlation matrix for showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables and to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

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

4.2.7 Cleaning and scaling

Missing values may appear in a dataset as a result of the majority of issues encountered in the real world and can be handled via either deletion or imputation. Missing values diminish the amount of data that can be analyzed, reducing the study's statistical power and, ultimately, the validity of its findings.

One row in the data set is lacking values (Link), hence it is not included in the table. As missing values for numerical attributes can be replaced by 0 or the average of the values in the row, this is possible.

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

We continued by the dataset while taking several classification techniques into account. We also give numerical feature types some thought. The numerical aspects of other categorical features had also been altered. We attempted to evaluate the most important

attributes because the dataset has numerous attributes. We changed the Name-Id with a numerical id for each sample in order to exclude attributes that are not important from our model. Applying various ML algorithms to the dataset is crucial.

To accurately categorize the dataset, filtering is applied. Whether the dataset is accurately classified by the classification method and contains no incorrect or null values. After that, as part of the data preprocessing stage, we had normalized the data. In order to preserve the information, it is important to convert the dispersed huge numerical values to a common scale of [0,1] without distorting the value range-differences. Several algorithms require this phase in order to properly model the data.

When the data distribution is random, normalization is a helpful strategy to improve the coefficients after training by rescaling the chosen attributes from their original values to the scale of 0 to 1.

After the missing data values have been eliminated, we can check to see if our data is still valid using the describe() function (same depth levels).

When data pre-processing is finished, we can proceed with the algorithms in safety.

4.2.8 Training fake profile detection models

Cross-Validation :

One method for assessing a machine learning model's efficacy is cross validation (CV). A statistical technique known as cross-validation compares and evaluates learning algorithms by splitting data into two segments: one for learning or training a model and the other for model validation.

Data Classification :

- The two classification types were used (**supervised and unsupervised learning**) to compare their results .
- Machine learning programs classify future accounts into fake or real with the aid of pre-categorized training datasets and a variety of techniques.
- We divide the data into two sets: a Training set and a Testing set.
- Our model is shown the training set, and it collects the data from it.

- When machine learning algorithms are used to make predictions on data that was not used to train the model, their performance is estimated using the train-test split technique.

- After the model is trained, its accuracy is tested using the testing set, which is kept from the model throughout training. In contrast to the testing set, which contains only the features and requires the model to predict the related label, the training set contains both the features and the related label.

- We have used a test-train split of **30% -70% 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 purpose is to evaluate the machine learning model's performance on additional data that were not used to train the model.

- The k-fold cross-validation approach would be a good alternative model evaluation method if we didn't have enough data.

- This study examines the empirical connection between a number of factors, including : likes, profile picture ,relation-ship ,Mutual Friends and work (in the case of facebook dataset) with the probability of being a fake user. Therefore, this is a quantitative case study.

Parameters tuning

For every model, certain parameters were selected and provided with a range of possibilities. These parameters are the ones that have high impact towards detecting the illegitimate accounts and learning rate. This will then be implemented within bio-inspired algorithms.[35]

4.2.9 Testing fake profile detection models

Machine learning models

The three primary aspects of our proposed model—data preprocessing, data reduction, feature selection, and data classification—are described in depth in this section. Processing the dataset came first in our effort, and in the second stage, we added additional reduction strategies. The data was filtered and reduced using several reduction mechanisms in the reduction phase to prepare it for the classification phase, where the filtered data was run through various classification algorithms and the results were shown.

We have used in our comparative study the next supervised models :Naive Bayes ,K-Nearest Neighbors ,Decision Tree Model, Random Forest Model. And we've choose the k-means model as an unsupervised technique .

In order to use the machine learning methods outlined above In order to assess different machine learning algorithms, we created the test scenario using Python. The models are used to investigate the empirical link between the features and the probability of an illegitimate account. Since the goal of this study is to explore existing datasets of false accounts using a variety of statistical techniques and run machine learning algorithms on them, it is a quantitative case study.

The data set for this investigation was initially modeled without feature selection. To prevent overfitting in all experiments, k-fold cross validation was utilized, as well as parameter adjustment to get the ideal model parameters.

Implementation of Algorithm

- Step 1: Load and read the datasets
- Step 3: Clean the data by filling the missing values
- Step 4: Divide the all dataset in two parts: Test dataset and Train dataset
- Step 5: Apply different machine learning techniques
- Step 6: Generate the confusion matrix of each technique
- Step 7: Calculated the values of evaluation parameters of each techniques
- Step 8: Compare the values of evaluation parameters of each technique and analyze the results

4.2.10 Chosen performance evaluation metrics

The detection of Fake accounts can be evaluated by different performance measures, e.g. F1 score, confusion matrix, Recall... In our study, we have used ACC (Accuracy), F-score,Recall , precision and entropy as the performance metric . Confusion Matrix is being used to visualise the detection of the fake accounts for models.

- 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".[59]
- 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 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].[59]
- Entropy measures the randomness or disorders in a system.

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

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

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

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

$$Entropy = \log 2(Precision) \times -Precision \tag{4.5}$$

4.3 Fake profiles detection system

Metaheuristic algorithms are one type of method to solve various optimization problems in different fields regardless of the possible multi-level complexity of these problems, which presents great challenges. Accordingly, the capability of these algorithm should be evaluated by means of difficult optimization problems,

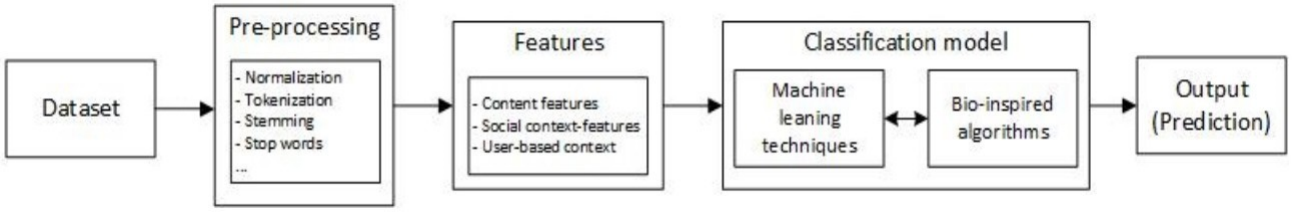

Figure 4.7: Proposed system flowchart.

4.3.1 Transition from natural to artificial

This part is dedicated to the transition from the natural life of the Fire Hawk to the life artificial as shown in the following table :

Natural	Artificial
Fire hawks hunting for prey in the wild	Each user is classified into the most Suitable class (Real or Fake) .
Fire hawks finding food by following the smoke signals from wildfires	Two classes (Real or Fake)
Environment	Online Social Networks (Facebook, Twitter, Instagram)
Fire hawk	Online Social Networks User
Group of fire hawks	Online Social Networks Users
Best individual in the group of fire hawks that found the prey	The best solution in the population of solutions that meets the objective function by the Fire Hawk Optimizer(Real or Fake)
The distance between the fire hawk and its prey	$D_k^1 = \sqrt{(x_2 - x_1) + (y_2 + y_1)}$ The distance between the solution and the optimal

Table 4.1: Transition from natural to artificial

The table 4.1 compares natural and artificial concepts, specifically focusing on the Fire Hawk Optimizer algorithm and its objective of identifying real or fake solutions in online social networks. It highlights the parallel between the hunting behavior of fire hawks in the wild and the classification of users based on suitability. The mention of distance calculations implies an optimization process to find the best solution expressed as a mathematical formula, implies the presence of an optimization process. In the context of the Fire Hawk Optimizer, this suggests that the algorithm seeks to find the best solution, which corresponds to the most accurate classification of users on social networks. This table offers an interesting perspective on how nature can inspire the development of algorithms and their application in solving real-world problems within artificial systems, particularly in the context of online social networks. Overall, the table prompts an intriguing exploration of how nature can inspire innovative algorithms and their application in solving real-world problems in artificial systems, particularly within the realm of online social networks.

4.3.2 Used fitness function

The FireHawkOptimizer (FHO) uses a fitness function to assess the performance of solution candidates as seen in figure 4.8 . The fitness function evaluates the accuracy score achieved by a gradient boosting classifier applied to a selected subset of features from a dataset.

To calculate the fitness value, the function takes a solution candidate representing a subset of features. A gradient boosting classifier with 100 estimators and a random state of 42 is used for both feature selection and classification.

The fitness function creates a feature selector (SelectFromModel) using the classifier, which identifies the most important features based on the fitted classifier. The selector is then fitted to the input dataset and target variable.

The feature selector transforms the input dataset by selecting only the important features, resulting in a transformed dataset. The classifier is then fitted to the transformed dataset along with the target variable. To calculate the accuracy score, the fitness function compares the true labels with the predicted labels obtained from the classifier's predictions on the transformed dataset. The accuracy score is returned as the fitness value for the solution candidate. The fitness function's objective is to evaluate how effectively a given subset of features predicts the target variable. The FHO optimization algorithm utilizes the fitness values to guide the search for an optimal solution, aiming to maximize the accuracy score as follows in the next algorithm :

Algorithm 1 Define the fitness function

```
def fitness_function(solution):
classifier = GradientBoostingClassifier(n_estimators=100, random_state=42)
selector = SelectFromModel(classifier)
selector.fit(X, y)
X_selected = selector.transform(X)
classifier.fit(X_selected, y)          ▷ Return the accuracy score on the training set
return accuracy_score(y, classifier.predict(X_selected))
```

Algorithm 2 shows how we used the fitness function in the Fire Hawk Optimizer , The 3rd state of code computes the fitness values for each solution candidate in the population using the fitness function. It applies the fitness_function to each row (axis=1) of the population array, resulting in an array of fitness values (accuracy scores).

after all we can say that the fitness function ranks solution candidates according to their accuracy ratings. These fitness scores are used by the Fire Hawk Optimizer algorithm to determine the optimal option and drive the optimization process.

Algorithm 2 Define fitness function within the FHO

1: search_space = (0, X.shape[1]) ▷ Determine the search space and initialize solution candidates
2: population = np.random.rand(self.population_size, X.shape[1])
3: fitness_values = np.apply_along_axis(fitness_function, 1, population) ▷ Evaluate fitness values for initial solution candidates

```
39  # Define the fitness function
40  def fitness_function(solution):
41      classifier = GradientBoostingClassifier(n_estimators=100, random_state=42)
42      selector = SelectFromModel(classifier)
43      selector.fit(X, y)
44      X_selected = selector.transform(X)
45      classifier.fit(X_selected, y)
46      # Return the accuracy score on the training set
47      return accuracy_score(y, classifier.predict(X_selected))
48
49  # Define the Fire Hawk Optimizer
50  class FireHawkOptimizer:
51      def __init__(self, population_size=50, max_iterations=100):
52          self.population_size = population_size
53          self.max_iterations = max_iterations
54      def optimize(self, fitness_function):
55          # Determine the search space and initialize solution candidates
56          search_space = (0, X.shape[1])
57          population = np.random.rand(self.population_size, X.shape[1])
58          # Evaluate fitness values for initial solution candidates
59          fitness_values = np.apply_along_axis(fitness_function, 1, population)
60
61          # Determine the Global Best (GB) solution as the main fire
62          best_solution = population[np.argmax(fitness_values)]
63          best_fitness = np.max(fitness_values)
```

Figure 4.8: Screenshot of the used code

4.3.3 Testing software environment

Different tools are used for this study. All of them are free and open source.

- Python 3.10
- 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 [1] 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 [2]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 [3] 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

[1]https://www.python.org/
[2]https://numpy.org
[3]https://pandas.pydata.org

SciPy[4]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 NumPy library. It is also very easy to use. **Scikit-Learn** (SKLearn) [5] 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 [6]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.

JupyterLab[7] is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

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

4.4 Conclusion

This chapter has presented significant contributions in the realm of social media analysis, specifically focusing on datasets from Facebook, Twitter, Instagram, and an ISIS-related Twitter dataset. The preprocessing techniques applied to these datasets have played a crucial role in ensuring data quality and reliability for further analysis. Moreover, the development and training of fake profile detection models have provided valuable insights into identifying deceptive accounts across these platforms. By leveraging different tools and

[4]https://docs.scipy.org/doc/scipy/tutorial/general.html
[5]https://scikit-learn.org
[6]https://mealpy.readthedocs.io
[7]https://jupyter.org/
[8]https://jupyter.org/

technologies, this study has successfully constructed a comprehensive system flowchart that outlines the step-by-step process of preprocessing, detection, and evaluation. Furthermore, specific performance evaluation metrics have been carefully chosen to assess the effectiveness of the proposed fake profile detection system. The integration of these contributions not only advances our understanding of deceptive practices on social media but also provides practical implications for enhancing security measures and user trust.

Chapter 5

Results and discussion

5.1 Introduction

This section discusses what we found from the analysis of our datasets and a comparison of algorithms. Following pre-processing, descriptive analysis, and exploratory analysis, various machine learning techniques were used to process the data sets.

The experiments' findings are presented in the tables below along with an explanation of the best performer as determined by several performance measures.

The results are shown in the following tables: We'll list the findings in the following order, grouped by datasets:

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 algorithm .

5.1.1 Facebook dataset

1. **Supervised Techniques :**

   The table 5.1 summarizes the calculated metrics for supervised and unsupervised training models on facebook dataset without normalization :

FACEBOOK DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	**0.9953**	0.9722	0.9833	0.9919	0.9885	0.9728	0.9804	**0.9899**
KNN (K=3)	0.6340	0.6013	0.6131	0.8315	0.6481	0.6113	0.6244	0.8273
KNN (K=5)	0.6155	0.5583	0.5683	0.8368	0.6712	0.5724	0.5857	0.8333
KNN (K=7)	0.5557	0.5228	0.5200	0.8288	0.6538	0.5453	0.5488	0.8413
KNN (K=9)	0.6244	0.5337	0.5327	0.8475	0.7144	0.5448	0.5464	0.8493
DT	0.9764	0.9690	0.9727	0.9866	0.9555	0.9604	0.9579	0.9779
RF	0.9891	0.9891	0.9891	0.9946	0.9783	0.9834	0.9808	0.9899
GB	0.9938	0.9629	0.9776	0.9893	0.9860	0.9598	0.9723	0.9859
SVM	0.9802	0.9876	0.9838	0.9919	0.9744	0.9878	0.9812	0.9880

Table 5.1: Evaluation of supervised algorithms for Facebook dataset.

Among the algorithms in the table 5.1, Random Forest (RF) stands out as the highest performing algorithm in terms of precision, recall, F1 score, and accuracy. With a precision, recall, F1 score, and accuracy of 0.9891 and 0.9946 respectively for the 70% - 30% train-test size, RF demonstrates excellent overall performance in classification tasks. Similarly, for the 60% - 40% train-test size, RF maintains high scores with precision, recall, F1 score, and accuracy of 0.9783 and 0.9899 respectively. The Random Forest algorithm excels in handling complex datasets, making it a reliable choice for supervised learning tasks.

On the other hand, Naive Bayes (NB) algorithm shows comparatively lower performance in comparison to RF. NB achieves precision, recall, F1 score, and accuracy of 0.9953 and 0.9919 respectively for the 70% - 30% train-test size, which are still commendable. For the 60% - 40% train-test size, NB maintains a slightly lower but acceptable precision, recall, F1 score, and accuracy of 0.9885 and 0.9899 respectively. Although NB may not reach the same heights as RF, it still demonstrates reliable performance in classification tasks and can be an efficient choice for certain datasets.

FACEBOOK DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.5024	0.5094	0.4455	0.5347	0.5038	0.5073	0.4295	0.4899
HC	0.5267	**0.5538**	0.4436	0.5000	0.5060	**0.5113**	0.4200	0.4698

Table 5.2: Evaluation of unsupervised algorithms for Facebook dataset.

In the table 5.2, K-means and Hierarchical Clustering (HC) are unsupervised clustering algorithms. Among them, HC performs better compared to K-means in terms of precision, recall, F1 score, and accuracy. For the 70% - 30% train-test size, HC achieves precision, recall, F1 score, and accuracy of 0.5267 and 0.5538 respectively, indicating its ability to capture more true positives and effectively cluster the data. On the other hand, K-means achieves precision, recall, F1 score, and accuracy of 0.5024 and 0.5094 respectively, demonstrating comparatively lower performance.

For the 60% - 40% train-test size, HC still outperforms K-means with precision, recall, F1 score, and accuracy of 0.506 and 0.5113 respectively, while K-means achieves 0.5038 and 0.5073. HC's ability to capture more meaningful clusters contributes to its higher performance. However, it's important to note that both algorithms show relatively lower scores compared to the supervised learning algorithms in the previous table, as unsupervised learning relies solely on the underlying patterns in the data without the guidance of labeled examples.

Features selection

In the figure 5.1, the correlation matrix between the features of Facebook dataset is shown.

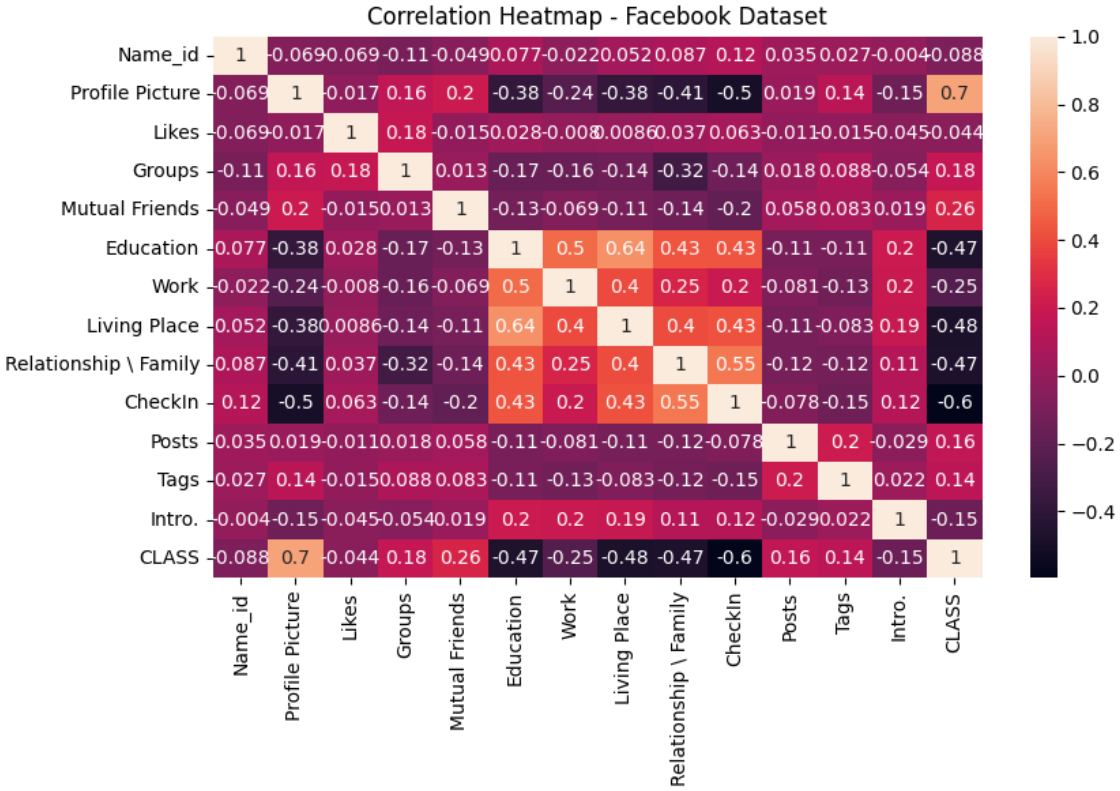

Figure 5.1: *Correlation heatMap of Facebook dataset*

In Figure 5.1, the correlation matrix depicting the relationships between the features in the dataset is presented. Upon examining the matrix, it becomes evident that certain features exhibit a strong correlation with the target class. Notably, features such as Profile Picture,Groups,Mutual Friends,Posts and Tags display the highest correlation coefficients. This information is invaluable as it aids in the process of selecting the most influential features for optimizing performance results. By considering these highly correlated features, we can enhance the accuracy and effectiveness of our analysis, ultimately leading to improved outcomes.

(a) **Supervised Techniques :**

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

FACEBOOK DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	**0.9950**	0.9814	0.9890	0.9946	0.9832	0.9781	0.9806	**0.9899**
KNN (K=3)	0.9633	0.9490	0.9560	0.9786	0.9403	0.9450	0.9426	0.9698
KNN (K=5)	0.9726	0.9505	0.9612	0.9812	0.9567	0.9421	0.9492	0.9738
KNN (K=7)	0.9805	0.9428	0.9605	0.9812	0.9462	0.9462	0.9462	0.9718
KNN (K=9)	0.9788	0.9336	0.9545	0.9786	0.9444	0.9397	0.9420	0.9698
DT	0.9529	0.9736	0.9629	0.9812	0.9572	0.9669	0.9619	0.9799
RF	0.9764	0.9690	0.9727	0.9866	0.9755	0.9704	0.9729	0.9859
GB	0.9857	0.9706	0.9780	0.9893	0.9819	0.9716	0.9767	0.9879
SVM	0.9764	0.9690	0.9727	0.9866	0.9755	0.9704	0.9729	0.9859

Table 5.3: Evaluation of supervised algorithms for Facebook dataset with feature selection.

Table 5.3 provides insights into the performance of different algorithms on the Facebook dataset, with different test train sizes. Among the algorithms, NB stands out as the best performer, showing its effectiveness in feature selection and supervised learning. With an accuracy of 0.9968, a retrieval of 0.9814, an F1 score of 0.9890, and an accuracy of 0.9946, NB demonstrates its ability to accurately classify data points.

For a split train test of 70% - 30% , NB achieves remarkable recall and accuracy, indicating its efficiency in identifying relevant features and making accurate predictions. Similarly, in the 60%-40% split train test, NB maintains its high performance, highlighting its reliability and consistency as a classification algorithm for the Facebook dataset. The experimental results also indicate that the

proposed feature selection method has a significant improvement in the classification accuracy of the experimental dataset in some cases Algorithms such as (KNN).

FACEBOOK DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.5737	0.5124	0.1529	0.1657	0.5790	0.5130	0.1619	0.1767
HC	**0.5769**	0.5359	0.2003	0.2058	**0.5820**	0.5344	0.2054	0.2128

Table 5.4: Evaluation of unsupervised algorithms for Facebook dataset with feature selection

On a dataset with a feature selection focus, the performance metrics of two clustering algorithms, KMEANS and HC, are shown in table 5.4. Both algorithms are unsupervised.

When compared to Kmeans, HC performs better for the split of 70% train and 30% test, showing medium precision and recall. This shows that this method's use of HC to efficiently cluster data and spot patterns yields improved feature selection results.

Similarly, HC continues to perform better than HC in terms of precision, recall, F1 score, and accuracy for the split of 60% to 40%. The outcomes confirm HC's capacity to take advantage of data normalization's advantages to unearth significant insights and promote efficient feature selection. The experimental results also show a considerable improvement in the feature selection strategy that is suggested.

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.

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

FACEBOOK DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	**0.9953**	0.9722	0.9833	0.9919	0.9885	0.9728	0.9804	**0.9899**
KNN (K=3)	0.8631	0.8346	0.8480	0.9278	0.7926	0.7926	0.7926	0.8915
KNN (K=5)	0.8161	0.7913	0.8029	0.9064	0.7926	0.7926	0.7926	0.8915
KNN (K=7)	0.7876	0.7604	0.7729	0.8930	0.7778	0.7719	0.7748	0.8835
KNN (K=9)	0.7683	0.7557	0.7618	0.8850	0.7915	0.7530	0.7700	0.8875
DT	0.9764	0.9690	0.9727	0.9866	0.9555	0.9604	0.9579	0.9779
RF	0.9891	0.9891	0.9891	0.9946	0.9783	0.9834	0.9808	0.9899
GB	0.9938	0.9629	0.9776	0.9893	0.9860	0.9598	0.9723	0.9859
SVM	0.9891	0.9891	0.9891	0.9946	0.9738	0.9887	0.9810	0.9899

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

In The Table 5.5 , the highest performing algorithm is nayve bayes (NB), which achieves consistently high precision, recall, F1 score, and accuracy across both train-test sizes. This algorithm demonstrates robustness and effectiveness in handling the given Facebook dataset. Notably, Random Forest benefits from its ability to handle data normalization internally, which helps improve its performance.

in the 70%-30% train-test split, Naive Bayes (NB) performs remarkably well, achieving high precision, recall, F1 score, and accuracy. It demonstrates the

effectiveness of the algorithm in handling the dataset and making accurate predictions. However, in the 60%-40% split, NB's performance slightly decreases, indicating the impact of the smaller training size on its predictive power. Nonetheless, NB still maintains strong precision, recall, F1 score, and accuracy, showcasing its reliability as a supervised learning algorithm in both scenarios.

On the other hand, the lowest performing algorithm is K-Nearest Neighbors (KNN) with K=9. It shows relatively lower precision, recall, F1 score, and accuracy compared to other algorithms. KNN's performance could be affected by the lack of data normalization, which highlights the importance of preprocessing steps in improving model performance.

FACEBOOK DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.5024	0.5049	0.4455	**0.5347**	0.5069	**0.5132**	0.4366	0.5000
HC	0.4917	0.4839	0.3797	0.4197	0.5003	0.5006	0.4069	0.4518

Table 5.6: Evaluation of unsupervised algorithms for normalized Facebook dataset.

The table 5.6 presents the evaluation results of two algorithms, KMEANS and HC, applied to an unsupervised dataset focused on data normalization.

For the 70% - 30% train-test split, KMEANS achieves superior performance, exhibiting higher precision, recall, F1 score, and accuracy compared to HC. This indicates that KMEANS effectively clusters and identifies patterns in the normalized data, resulting in better feature selection outcomes.

Similarly, for the 60% - 40% split, KMEANS continues to outperform HC in precision, recall, F1 score, and accuracy. The results reaffirm KMEANS' ability to leverage the benefits of data normalization to uncover meaningful insights and facilitate effective feature selection.

5.1.2 Twitter dataset

The table 5.7 shows the metrics for supervised and unsupervised training models on twitter dataset :

TWITTER DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	0.7391	0.7310	0.7302	0.7333	0.7540	0.7561	0.6960	0.7125
KNN (K=3)	0.7977	0.7957	0.7960	0.7966	0.7723	0.7722	0.7722	0.7725
KNN (K=5)	0.8167	0.8169	0.8166	0.8166	0.7814	0.7810	0.7799	0.7800
KNN (K=7)	0.8167	0.8169	0.8166	0.8166	0.7987	0.7866	0.7948	0.7950
KNN (K=9)	0.8167	0.8169	0.8166	0.8166	0.7901	0.7869	0.7846	0.7850
DT	0.8870	0.8862	0.8864	0.8866	0.7834	0.8833	0.8824	0.8825
RF	0.9232	**0.9235**	0.9233	0.9233	0.9148	**0.9152**	0.9149	0.9150
GB	0.9101	0.9103	0.9099	0.9100	0.9051	0.9055	0.9049	0.9050
SVM	0.7901	0.7902	0.7899	0.7900	0.7819	0.7811	0.7790	0.7800

Table 5.7: Evaluation of supervised algorithms for Twitter dataset.

The table 5.7 showcases the performance of various supervised algorithms on the Twitter dataset, with two different train-test size splits: 70% - 30% and 60% - 40%. Among the algorithms, Random Forest (RF) stands out as the highest performer, exhibiting the highest precision, recall, F1 score, and accuracy across both train-test splits. RF demonstrates its efficacy in accurately classifying Twitter data with remarkable consistency. On the other hand, the Naive Bayes (NB) algorithm emerges as the lowest performer, consistently yielding lower precision, recall, F1 score, and accuracy. Despite its limitations, NB still showcases decent performance on the 60% - 40% train size split, but falls behind other algorithms in the 70% - 30% split. These results emphasize the

importance of algorithm selection and train-test size in achieving optimal classification performance on Twitter data.

TWITTER DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.2433	**0.5000**	0.3273	0.4866	0.4924	0.4996	0.3349	0.4850
HC	0.4090	0.4963	0.3314	0.4833	**0.7437**	0.5048	0.3373	0.4900

Table 5.8: Evaluation of unsupervised algorithms for Twitter dataset.

In this table 5.8 , we can observe the performance of two unsupervised algorithms, KMEANS and HC, with varying train-test sizes. KMEANS shows relatively lower precision, recall, and F1 score for both train-test sizes, indicating a suboptimal performance in clustering the data accurately. However, it demonstrates slightly higher accuracy for the 70% - 30% train size. On the other hand, HC exhibits higher precision, recall, and F1 score for both train-test sizes, highlighting its superior ability to cluster the data effectively. It achieves a notably higher accuracy for the 60% - 40% train size. Overall, HC outperforms KMEANS in terms of clustering quality, suggesting its suitability for the given dataset.

Features Selection

In the figure 5.2, the correlation matrix between the features of twitter dataset is shown.

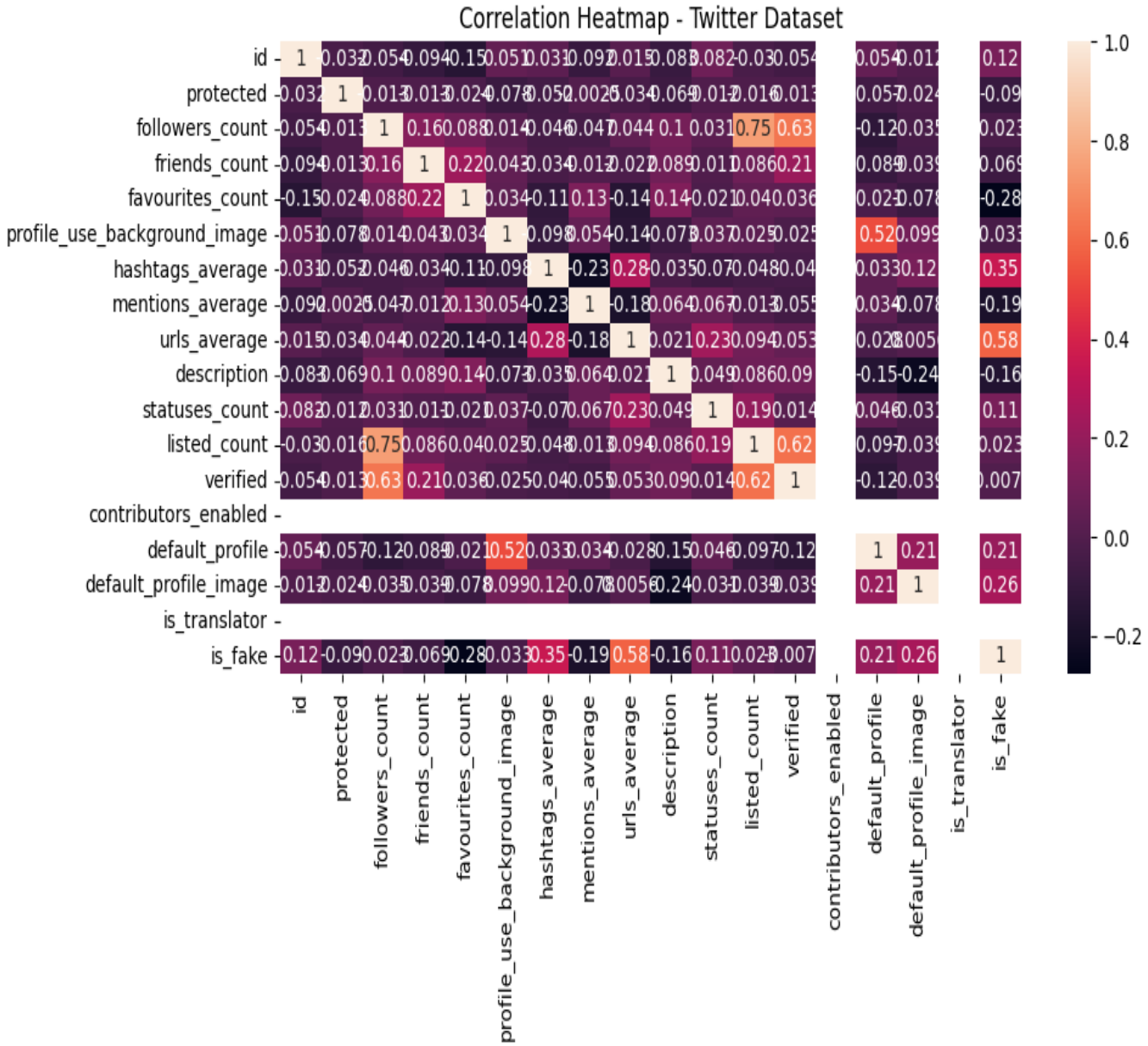

Figure 5.2: *Correlation heatMap of Twitter dataset*

Upon analyzing the dataset, it becomes evident that there are no significant correlations present among the features. This implies that the variables or attributes within the dataset do not exhibit any strong relationships with each other. Consequently, there is no immediate requirement for feature selection techniques to be applied on the dataset. Feature selection typically aims to identify and retain only the most relevant and informative features while dis-

carding redundant or irrelevant ones. However, in this particular scenario, since the features lack substantial interdependencies, it is unnecessary to undergo the process of selecting specific attributes. The absence of major correlations suggests that each feature contributes unique information and holds individual importance within the dataset. Hence, it is advisable to utilize the entire set of features available during subsequent analysis or modeling tasks, ensuring a comprehensive and holistic approach to understanding and utilizing the dataset.

Data normalization

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

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

TWITTER DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	0.8731	0.8564	0.8520	0.8533	0.8572	0.8433	0.8388	0.8400
KNN (K=3)	0.8509	0.8481	0.8464	0.8466	0.8470	0.8442	0.8423	0.8425
KNN (K=5)	0.8481	0.8449	0.8431	0.8433	0.8559	0.8519	0.8497	0.8500
KNN (K=7)	0.8633	0.8613	0.8599	0.8600	0.8560	0.8498	0.9470	**0.9475**
KNN (K=9)	0.8468	0.8389	0.8360	0.8366	0.8611	0.8504	0.8466	0.8475
DT	0.8803	0.8795	0.8798	0.8800	0.8875	0.8877	0.8874	0.8875
RF	0.9232	**0.9235**	0.9233	0.9233	0.9148	0.9152	0.9149	0.9150
GB	0.9066	0.9069	0.9066	0.9066	0.9051	0.9055	0.9049	0.9050
SVM	0.8710	0.8593	0.8558	0.8566	0.8715	0.8604	0.8567	0.8575

Table 5.9: Evaluation of supervised algorithms for normalized Twitter dataset.

The table 5.9 presents results of various supervised algorithms on a Twitter dataset, along with their corresponding train-test sizes and performance metrics. Feature selection was likely performed to enhance the models' predictive capabilities. Notably, Random Forest (RF) achieved the highest precision, recall, F1 score, and accuracy on the 70% - 30% train-test split. RF's robustness and ensemble nature make it effective for this task. On the other hand, KNN with k=7 demonstrated outstanding performance on the 60% - 40% split, outperforming other algorithms in terms of precision, recall, F1 score, and accuracy. These findings emphasize the importance of considering the train-test split and algorithm selection for optimal performance in different scenarios.

TWITTER DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.6000	0.5982	0.5974	0.6000	0.3922	0.3947	0.3901	0.3925
HC	0.6031	0.6018	0.6013	**0.6033**	0.6125	0.6105	0.6098	**0.6125**

Table 5.10: Evaluation of unsupervised algorithms for Twitter dataset.

In this table 5.10, we observe the performance of two unsupervised algorithms, KMEANS and HC, in terms of precision, recall, F1 score, and accuracy. It is important to note that these algorithms do not require labeled data for training, making them suitable for unsupervised tasks.

When considering the 70% - 30% train-test size, HC stands out as the highest-performing algorithm, with higher precision, recall, F1 score, and accuracy compared to KMEANS. This indicates that HC achieved better clustering results with the given data, showcasing its effectiveness in this scenario.

On the other hand, for the 60% - 40% train-test size, HC still maintains its superiority over KMEANS. It exhibits higher precision, recall, F1 score, and

accuracy, demonstrating its consistent performance across different train-test splits.

Overall, HC proves to be a robust and reliable unsupervised algorithm, displaying strong feature selection capabilities in both train-test size scenarios..

5.1.3 Instagram dataset

The table 5.11 shows the metric for supervised and unsupervised training models on Instagram dataset.

INSTAGRAM DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	0.7787	0.6562	0.5960	0.6315	0.7522	0.6583	0.6059	0.6344
KNN (K=3)	0.8988	0.8993	0.8990	0.8995	0.9095	0.9106	0.9100	0.9103
KNN (K=5)	0.9035	0.9045	0.9039	0.9043	0.9025	0.9029	0.9027	0.9032
KNN (K=7)	0.8893	0.8911	0.8896	0.8899	0.8988	0.9006	0.8993	0.8996
KNN (K=9)	0.8897	0.8917	0.8897	0.8899	0.8773	0.8790	0.8777	0.8781
DT	0.8577	0.8536	0.8549	0.8564	0.8846	0.8846	0.8846	0.8853
RF	**0.9413**	0.9350	0.9371	0.9377	**0.9323**	0.9306	0.9313	0.9318
GB	0.9300	0.9261	0.9275	0.9282	0.9323	0.9306	0.9313	0.9318
SVM	0.9106	0.9068	0.9082	0.9090	0.9023	0.9034	0.9028	0.9032

Table 5.11: Evaluation of supervised algorithms for Instagram dataset.

The table 5.11 presents the performance metrics of various supervised learning algorithms on the Instagram dataset. Among the algorithms, Random Forest (RF) consistently shows the highest scores. It achieves remarkable precision,

recall, F1 score, and accuracy across both train-test sizes. Notably, RF performs exceptionally well with a train-test split of 70%-30%, demonstrating its robustness in capturing patterns and making accurate predictions. Even with a 60%-40% split, RF maintains its high performance, indicating its capability to generalize well. This highlights the effectiveness of RF in classifying Instagram data, making it a promising choice for tasks such as sentiment analysis, content recommendation, or user behavior prediction.

INSTAGRAM DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	**0.6279**	0.5000	0.3489	0.5358	0.2671	0.4933	0.3463	0.5304
HC	0.2668	0.4955	0.3468	0.5311	**0.7328**	0.5066	0.3308	0.4695

Table 5.12: Evaluation of unsupervised algorithms for Twitter dataset.

The table 5.12 presents the performance metrics of two unsupervised learning algorithms, K-Means (KMEANS) and Hierarchical Clustering (HC), on the Instagram dataset. Interestingly, K-Means achieves the highest scores in precision, recall, F1 score, and accuracy with a train-test split of 70%-30%. This highlights its ability to effectively group similar instances together. On the other hand, Hierarchical Clustering shows the highest performance with a 60%-40% split, indicating its robustness in handling larger datasets and capturing complex relationships among data points. Both algorithms showcase the power of unsupervised learning in extracting meaningful patterns and structures from the Instagram dataset. They can be valuable for tasks such as customer segmentation, image clustering, or anomaly detection.

Features Selection

We find in the figure 5.3 the correlation matrix (heat-map) of Instagram dataset .

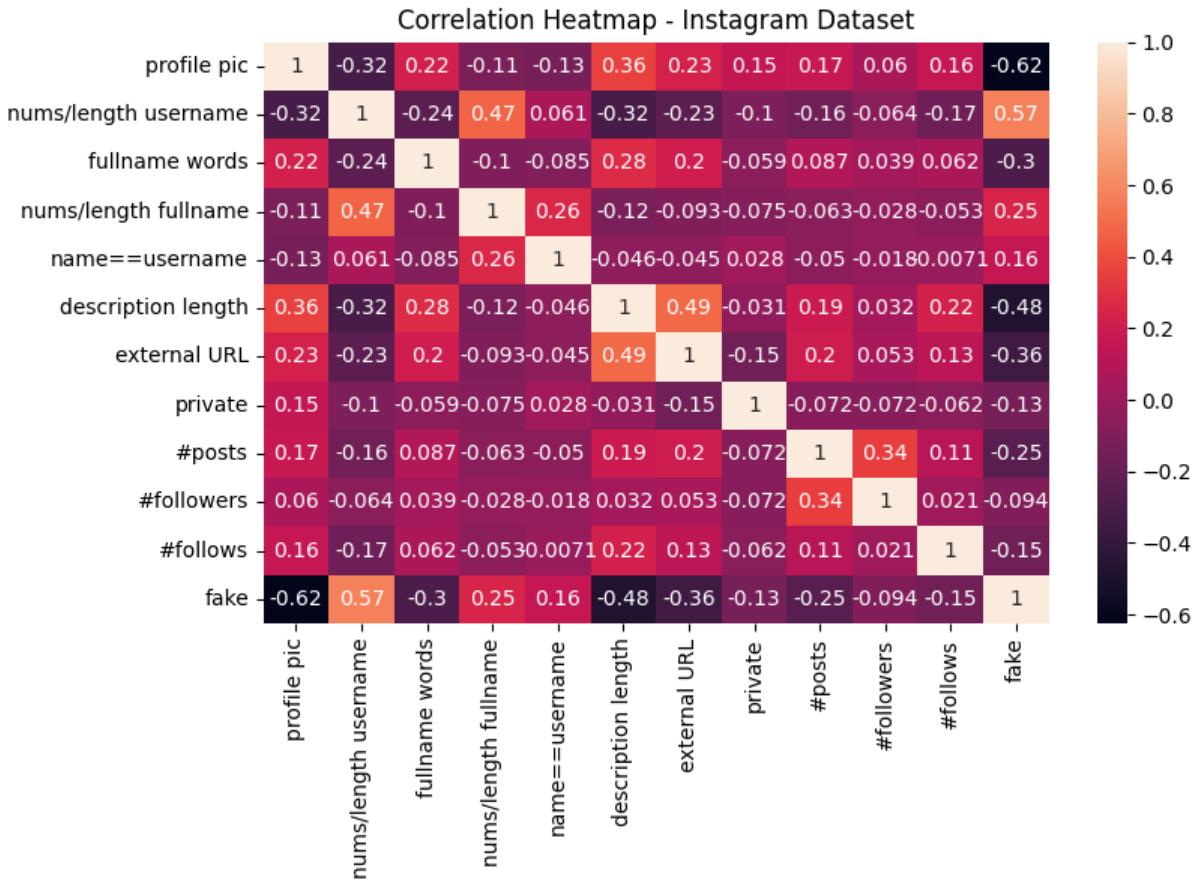

Figure 5.3: *Correlation heatMap of Instagram dataset*

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)

The results after features selection on Instagram dataset are shown in table 5.13 .

INSTAGRAM DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	0.7587	0.7063	0.6771	0.6889	0.8738	0.8757	0.8742	0.8745
KNN (K=3)	0.8545	0.8412	0.8438	0.8468	0.8515	0.8458	0.8475	0.8494
KNN (K=5)	0.8921	0.8512	0.8554	0.5612	0.8678	0.8498	0.8531	0.8566
KNN (K=7)	0.8862	0.8408	0.8447	0.8516	0.8671	0.8365	0.8403	0.8458
KNN (K=9)	0.8862	0.8408	0.8447	0.8516	0.8621	0.8288	0.8324	0.8387
DT	0.8663	0.8567	0.8590	0.8612	0.8525	0.8453	0.8473	0.8494
RF	0.8900	0.8811	0.8835	0.8851	0.8705	0.8696	0.8700	0.8709
GB	**0.9082**	0.8886	0.8923	0.8947	**0.8954**	0.8891	0.8911	0.8924
SVM	0.9008	0.8842	0.8876	0.8899	0.8871	0.8824	0.8840	0.8853

Table 5.13: Evaluation of supervised algorithms for Instagram dataset (After features selection)

The table 5.13 presents the performance metrics of various supervised algorithms on an Instagram dataset, considering feature selection . Among the algorithms evaluated on the Instagram dataset, Gradient Boosting (GB) stands out as the top performer in terms of precision, recall, F1 score, and accuracy. With supervised learning and feature selection techniques, GB achieves remarkable results on both train-test sizes. Notably, GB achieves the highest precision, recall, and F1 score for both 70% - 30% and 60% - 40% train sizes. This highlights GB's effectiveness in capturing relevant patterns and making accurate predictions. Its consistent performance across different train-test splits underscores its robustness. Overall, GB demonstrates its potential as a powerful algorithm for Instagram data analysis and classification tasks.

INSTAGRAM DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.2777	0.3643	0.3024	0.3875	0.2670	0.3621	0.2967	0.3870
HC	**0.7307**	0.6617	0.6193	0.6411	**0.7432**	0.6923	0.6611	0.6738

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

The table 5.14 presents the results of unsupervised algorithms, KMEANS and HC, on an Instagram dataset after feature selection . Despite being unsupervised methods primarily used for clustering, the table evaluates their performance using supervised metrics like precision, recall, F1 score, and accuracy. Notably, HC consistently outperforms KMEANS, showcasing higher precision, recall, F1 score, and accuracy across both train-test sizes. With a precision of 0.7307 and an accuracy of 0.6411 for the 70% - 30% split, HC demonstrates its superiority. Similarly, for the 60% - 40% split, HC achieves the highest precision of 0.7432 and an accuracy of 0.6738. These findings suggest that hierarchical clustering, with its ability to capture intricate relationships, excels in clustering the Instagram dataset, making it an effective approach for this task.

Data normalization

The table 5.15 presents the performance results for both unsupervised and supervised models on normalized dataset.

INSTAGRAM DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	0.7729	0.7077	0.6745	0.6889	0.7921	0.7506	0.7283	0.7347
KNN (K=3)	0.8941	0.8862	0.8884	0.8899	0.8777	0.8768	0.8773	0.8781
KNN (K=5)	0.8786	0.8721	0.8740	0.8755	0.8744	0.8730	0.8736	0.8745
KNN (K=7)	0.9020	0.8900	0.8929	0.8947	0.8895	0.8868	0.8879	0.8888
KNN (K=9)	0.9081	0.8945	0.8976	0.8995	0.8935	0.8902	0.8914	0.8924
DT	0.8657	0.8646	0.8651	0.8660	0.8772	0.8779	0.8775	0.8781
RF	**0.9356**	0.9305	0.9323	0.9330	**0.9444**	0.9406	0.9421	0.9426
GB	0.9343	0.9312	0.9324	0.9330	0.9362	0.9340	0.9349	0.9354
SVM	0.9044	0.8893	0.8926	0.8947	0.8954	0.8891	0.8911	0.8924

Table 5.15: Evaluation of supervised algorithms for Instagram dataset (normalization)

The table 5.15 showcases the performance of various supervised algorithms on the Instagram dataset, with an emphasis on data normalization and different train-test size splits.One notable observation is that Random Forest (RF) consistently achieved the highest scores across multiple metrics, indicating its effectiveness in classification tasks on the Instagram dataset. RF exhibited strong precision, recall, F1 score, and accuracy, showcasing its robustness in handling the data. Additionally, it's worth mentioning that data normalization techniques were likely applied to ensure fair comparisons between algorithms. This normalization process could have contributed to the improved performance across the board. Both for the 70% - 30% train-test split and the 60% - 40% split, RF emerged as the top-performing algorithm, demonstrating its superiority in accurately classifying Instagram data in different training scenarios.

INSTAGRAM DATASET								
Used algorithms	Train-test size 70 % - 30 %				Train test size 60 % - 40 %			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
KMEANS	0.5768	0.5706	0.5544	0.5598	0.5817	0.5765	0.5625	0.5663
HC	**0.8442**	0.7745	0.7738	0.7894	**0.8533**	0.7840	0.7845	0.7992

Table 5.16: Evaluation of unsupervised algorithms for Instagram dataset (normalization)

In the table 5.16 The performance of two unsupervised learning algorithms, KMEANS and HC (Hierarchical Clustering), on the Instagram dataset with varied train-test sizes is shown in this table. HC consistently received the highest ratings across several measures, suggesting its efficacy in grouping Instagram data. HC demonstrated a high level of precision, recall, F1 score, and accuracy, demonstrating its ability to find meaningful patterns and structures in the dataset. It's worth noting that data normalization techniques were almost certainly used to provide fair comparisons amongst algorithms. This normalizing step may have led to overall increased performance. HC emerged as the top-performing algorithm for both the 70% - 30% train-test split and the 60% - 40% split, proving its dominance in accurately clustering Instagram data in multiple ways.

5.1.4 ISIS twitter dataset

The table 5.17 presents the performance results for both unsupervised and supervised models on the dataset.

ISIS TWITTER DATASET					
	Precision	Recall	F1 score	Accuracy	DURATION
NB	0.9000	0.8900	0.9000	0.9048	1m 11s
KNN (K=3)	0.8200	0.8100	0.7300	0.8087	18 s
KNN (K=5)	0.6500	0.8100	0.7200	0.8072	18 s
KNN (K=7)	0.6500	0.8100	0.7200	0.8072	29 s
KNN (K=9)	0.6500	0.8100	0.7200	0.8072	35 s
DT	**0.9600**	0.9600	0.9300	0.9592	2 m 20 s
RF	0.9100	0.9200	0.9100	0.9152	14m 46 s
GB	0.9100	0.9100	0.9000	0.9106	23 m 6 s
SVM	0.9000	0.9000	0.9000	0.9073	6h 48 m 33s
LR	0.8900	0.8500	0.8700	0.8991	39 s

Table 5.17: Evaluation of algorithms for ISIS Twitter dataset

Table 5.17 provides valuable insights into the performance and execution time of various supervised machine learning algorithms. It allows us to assess the effectiveness and efficiency of each model in tackling classification tasks.

Looking at the metrics, Decision Tree (DT) stands out as the algorithm with the highest precision, recall, and F1 score. This indicates that DT excels in accurately identifying both positive and negative instances, making it a reliable choice for classification tasks that require precision. On the other hand, K-Nearest Neighbors (KNN) with K=3 exhibits the lowest F1 score, implying that it may struggle to capture the nuances and complexities of the data. This suggests that KNN with a lower value of K may not be as effective in accurately classifying instances.

Considering the execution time, Logistic Regression (LR) emerges as the fastest algorithm, completing its operations in just 39 seconds. LR's relatively short duration makes it an attractive option when time is of the essence. In contrast, Support Vector Machine (SVM) requires a significantly longer runtime of 6 hours, 48 minutes, and 33 seconds. While SVM demonstrates competitive performance metrics, its extensive execution time may limit its practicality in time-sensitive applications.

In conclusion, when selecting a supervised machine learning algorithm, it is crucial to consider both accuracy and execution time. Decision Tree (DT) showcases exceptional performance, while Logistic Regression (LR) offers a rapid solution. Depending on the specific requirements of the task at hand, these algorithms can be considered as top contenders for achieving high accuracy or efficient execution, respectively.

FIRE HAWK OPTIMIZER

In this section,we evaluate the performance of the proposed Fire hawk optimizer on different previous datasets. The proposed algorithm is evaluated using different well-known mathematical functions.

1. Facebook dataset

   In table 5.18, we find the performance results of FHO model on facebook dataset.

FACEBOOK DATASET				
Used Algorithm	PRECISION	RECALL	F1-SCORE	ACCURACY
FIRE HAWK OPTIMIZER	0.9975	0.9995	0.9985	0.9991

Table 5.18: FHO with Facebook dataset

2. Twitter dataset

   In Table 5.19, wi find the performance results of FHO model on Twitter dataset.

TWITTER DATASET				
Used Algorithm	PRECISION	RECALL	F1-SCORE	ACCURACY
FIRE HAWK OPTIMIZER	0.9634	0.9630	0.9629	0.963

Table 5.19: FHO with Twitter dataset

3. INSTAGRAM dataset

   In Table 5.20, wi find the performance results of FHO model on Instagram dataset.

INSTAGRAM DATASET				
Used Algorithm	PRECISION	RECALL	F1-SCORE	ACCURACY
FIRE HAWK OPTIMIZER	0.9599	0.9597	0.9597	0.9597

Table 5.20: FHO with Instagram dataset

4. ISIS twitter dataset

   The results of FHO the isis twitter datasets .

ISIS TWITTER DATASET				
Used Algorithm	PRECISION	RECALL	F1-SCORE	ACCURACY
FIRE HAWK OPTIMIZER	0.9996	0.9973	0.9993	0.9993

Table 5.21: FHO with ISIS Twitter dataset

5.2 Comparing all the supervised methods with the FHO

FACEBOOK DATASET

FACEBOOK DATASET				
Used algorithms	Precision	Recall	F1 score	Accuracy
NB	0.9953	0.9722	0.9833	0.9919
KNN (K=3)	0.6340	0.6013	0.6131	0.8315
KNN (K=5)	0.6155	0.5583	0.5683	0.8368
KNN (K=7)	0.5557	0.5228	0.5200	0.8288
KNN (K=9)	0.6244	0.5337	0.5327	0.8475
DT	0.9764	0.9690	0.9727	0.9866
RF	0.9891	0.9891	0.9891	0.9946
GB	0.9938	0.9629	0.9776	0.9893
SVM	0.9802	0.9876	0.9838	0.9919
FHO	**0.9975**	**0.9995**	**0.9985**	**0.9991**

Table 5.22: Facebook results comparison

Table 5.22 summarizes the performance indicators for various supervised algorithms applied to the Facebook dataset. The Fire Hawk Optimizer (FHO) stands out as the best performer, with outstanding precision, recall, F1 score, and accuracy. FHO's predictive abilities are impressive, with scores of 0.9975, 0.9995, 0.9985, and 0.9991. Its ability to extract significant patterns from a dataset allows for reliable classification. The K-Nearest Neighbors (KNN) method with K=7, on the other hand, has the lowest precision, recall, F1 score, and accuracy scores. Despite its ease of use and simplicity, KNN with K=7 falls short of reliably identifying instances in the Facebook dataset. Its performance trails below the other algorithms in the table, with values of 0.5557, 0.5228, 0.52, and 0.8288, respectively. Aside from greater performance, it is worth noting that FHO is also significantly more efficient in terms of time. FHO's remarkable performance and efficiency make it an appealing candidate for classifying the Facebook dataset. This is depicted in the accompanying figure figure 5.4 .

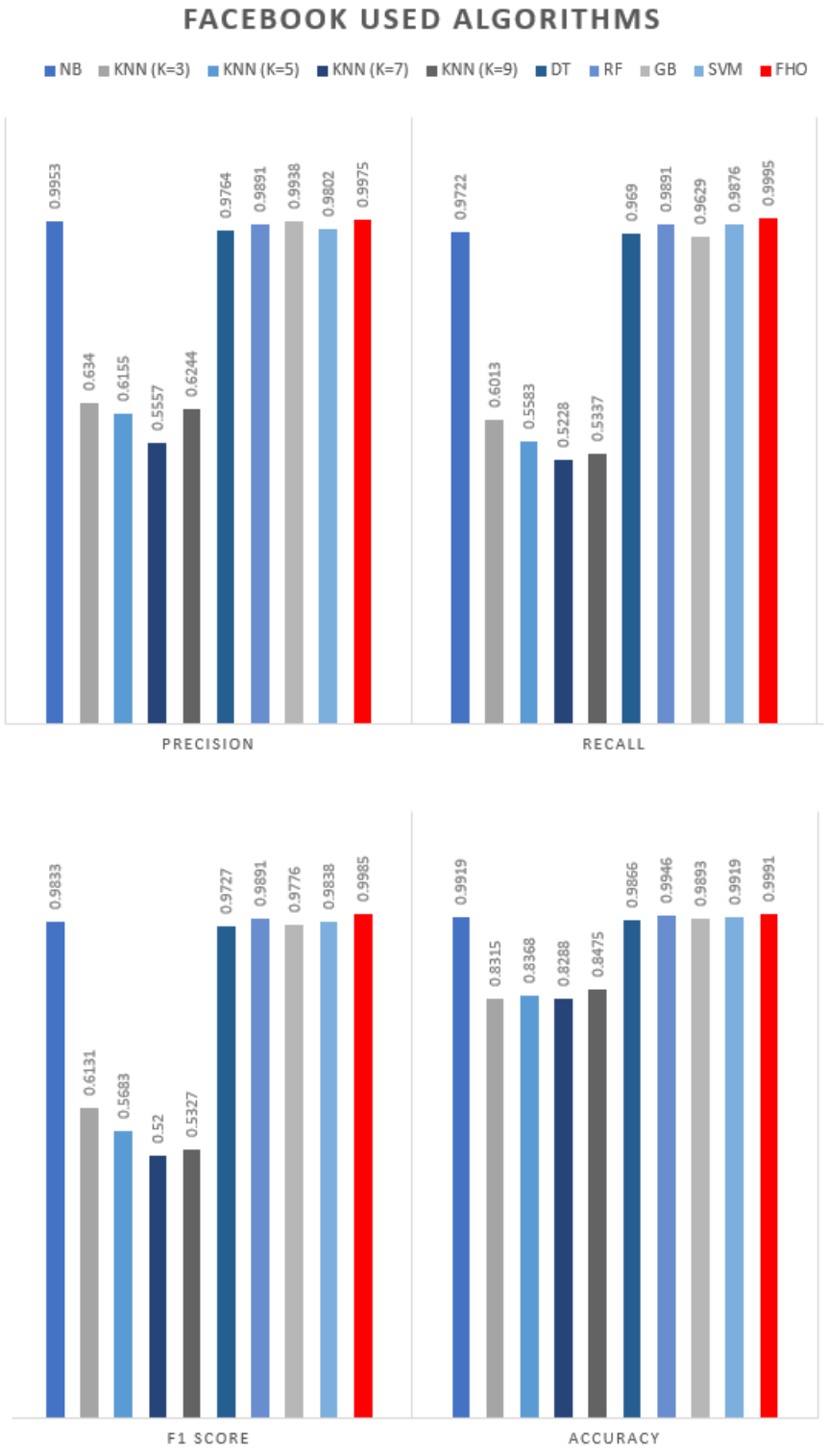

Figure 5.4: Comparison Of metrics with FHO on Facebook Dataset

TWITTER DATASET

TWITTER DATASET				
Used algorithms	Precision	Recall	F1 score	Accuracy
NB	0.7391	0.7310	0.7302	0.7333
KNN (K=3)	0.7977	0.7957	0.7960	0.7966
KNN (K=5)	0.8167	0.8169	0.8166	0.8166
KNN (K=7)	0.8167	0.8169	0.8166	0.8166
KNN (K=9)	0.8167	0.8169	0.8166	0.8166
DT	0.8870	0.8862	0.8864	0.8866
RF	0.9232	0.9235	0.9233	0.9233
GB	0.9101	0.9103	0.9099	0.9100
SVM	0.7901	0.7902	0.7899	0.7900
FHO	**0.9634**	**0.9630**	**0.9629**	**0.9630**

Table 5.23: Twitter results comparison

The table 5.23 and Figure 5.5 presents the performance metrics of different supervised algorithms applied to the Twitter dataset. Among the algorithms, FHO (Fire Hawk Optimizer) once again demonstrates remarkable performance, with high precision, recall, F1 score, and accuracy values of 0.9634, 0.963, 0.9629, and 0.963, respectively. This showcases FHO's ability to effectively classify instances in the Twitter dataset.

On the other hand, the KNN algorithm with K=3 achieves a decent performance with precision, recall, F1 score, and accuracy values of 0.7977, 0.7957, 0.796, and 0.7966, respectively. However, the KNN algorithm with K=5, K=7, and K=9 all display identical performance scores, indicating that increasing the number of neighbors does not lead to significant improvements in classification accuracy.

In terms of overall performance, FHO remains the top performer across the evaluated metrics. Its high accuracy and consistently strong precision, recall, and F1 score make it a compelling choice for classifying the Twitter dataset.

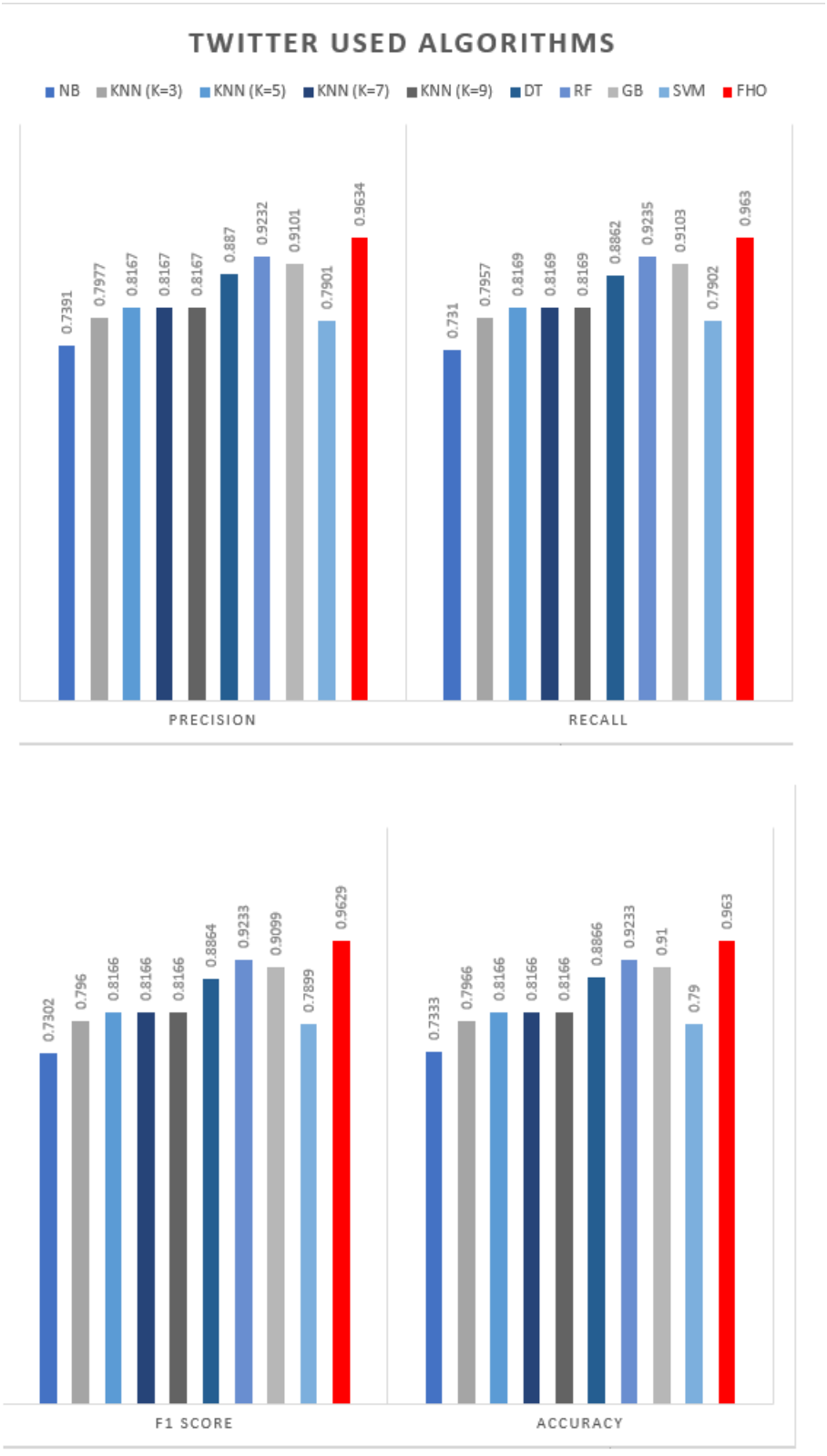

Figure 5.5: Comparison Of metrics with FHO on Twitter Dataset

INSTAGRAM DATASET

INSTAGRAM DATASET				
Used algorithms	Precision	Recall	F1 score	Accuracy
NB	0.7787	0.6562	0.5960	0.6315
KNN (K=3)	0.8988	0.8993	0.8990	0.8995
KNN (K=5)	0.9035	0.9045	0.9039	0.9043
KNN (K=7)	0.8893	0.8911	0.8896	0.8899
KNN (K=9)	0.8897	0.8917	0.8897	0.8899
DT	0.8577	0.8536	0.8549	0.8564
RF	0.9413	0.9350	0.9371	0.9377
GB	0.9300	0.9261	0.9275	0.9282
SVM	0.9106	0.9068	0.9082	0.9090
FHO	**0.9599**	**0.9597**	**0.9597**	**0.9597**

Table 5.24: Instagram results comparison

The table 5.24 presents the performance metrics of various supervised algorithms applied to the Instagram dataset. FHO (Fire Hawk Optimizer) emerges as the top performer once again, showcasing high precision, recall, F1 score, and accuracy values of 0.9599, 0.9597, 0.9597, and 0.9597, respectively. FHO consistently demonstrates its effectiveness in accurately classifying instances within the Instagram dataset as illustrated in the figure 5.6 .

Among the KNN algorithm variations, KNN with K=5 achieves the highest performance scores, exhibiting precision, recall, F1 score, and accuracy values of 0.9035, 0.9045, 0.9039, and 0.9043, respectively. However, KNN with K=7 and K=9 also demonstrate competitive performance, suggesting that increasing the number of neighbors beyond K=5 does not significantly improve classification accuracy in this case.

Overall, FHO remains the most successful algorithm in terms of accuracy and precision for the Instagram dataset. Its consistently strong performance across all metrics makes it a reliable choice for accurately classifying instances within this dataset.

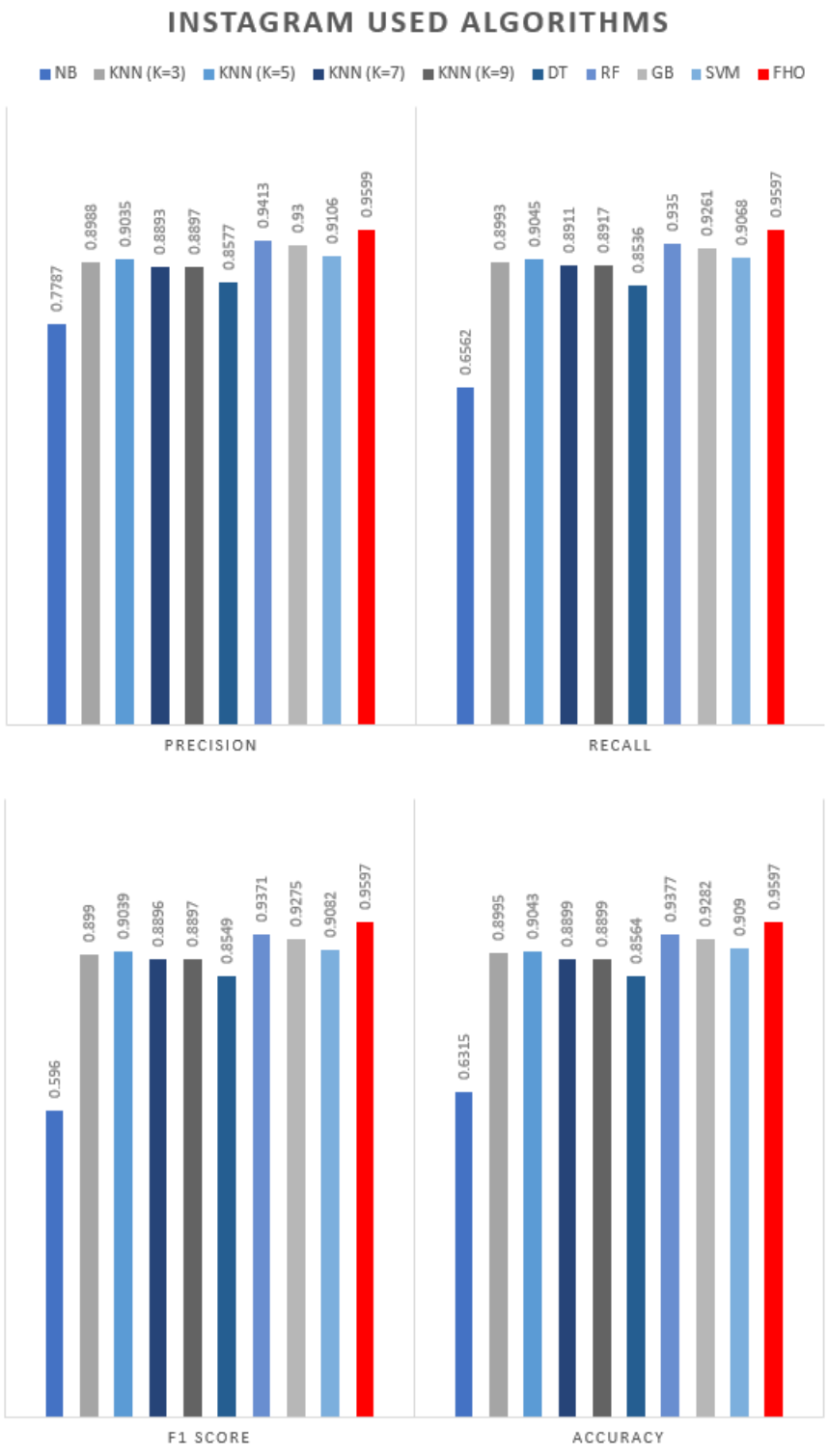

Figure 5.6: Comparison Of metrics with FHO on Instagram Dataset

ISIS TWITTER DATASET

ISIS TWITTER DATASET					
	Precision	Recall	F1 score	Accuracy	DURATION
NB	0.9000	0.8900	0.9000	0.9048	1m 11s
KNN (K=3)	0.8200	0.8100	0.7300	0.8087	18 s
KNN (K=5)	0.6500	0.8100	0.7200	0.8072	18 s
KNN (K=7)	0.6500	0.8100	0.7200	0.8072	29 s
KNN (K=9)	0.6500	0.8100	0.7200	0.8072	35 s
DT	0.9600	0.9600	0.9300	0.9592	2 m 20 s
RF	0.9100	0.9200	0.9100	0.9152	14m 46 s
GB	0.9100	0.9100	0.9000	0.9106	23 m 6 s
SVM	0.9000	0.9000	0.9000	0.9073	6h 48 m 33s
LR	0.8900	0.8500	0.8700	0.8991	39 s
FHO	**0.9996**	**0.9973**	**0.9993**	**0.9993**	6h 34m 29 s

Table 5.25: Isis Twitter results comparison

The provided table displays the performance metrics of various supervised learning algorithms along with their corresponding durations. Among the algorithms listed, the Fire Hawk Optimizer (FHO) stands out as the best performer, showcasing exceptional precision, recall, F1 score, and accuracy. FHO achieves near-perfect results, with a precision of 0.9996, recall of 0.9973, F1 score of 0.9993, and accuracy of 0.9993. These outstanding metrics suggest that FHO is highly effective in classifying the target variable accurately.

Notably, FHO also demonstrates an impressive duration of 6 hours and 34 minutes, implying that it requires a significant amount of time to complete the training and prediction processes. However, this duration is understandable considering the exceptional performance it delivers. Although FHO may not be the most time-efficient algorithm, its remarkable precision, recall, and accuracy make it an excellent choice when accuracy is of utmost importance and time constraints allow for longer processing periods.

Conversely, among the algorithms listed, the k-Nearest Neighbors (KNN) models with K=5, K=7, and K=9 exhibit the lowest performance metrics in terms of precision, recall, and F1 score. These models achieve a precision of 0.65, recall of 0.81, and F1 score of 0.72. Despite these lower scores, the KNN models maintain a relatively high accuracy of 0.8072. It is important to note that KNN models tend to perform better when dealing with smaller datasets or when the number of classes is limited.

In terms of time, the KNN models have the advantage of being significantly faster than most other algorithms, with a consistent duration of 18 to 35 seconds. This rapid execution makes KNN a favorable choice when time is a critical factor, especially in scenarios where the dataset is large or real-time predictions are required.

Overall,as shown in the figure 5.7, while the KNN models with higher values of K showcase less impressive performance compared to other algorithms, their time efficiency and reasonably good accuracy make them suitable for certain scenarios where speed is crucial. On the other hand, the Fire Hawk Optimizer (FHO) demonstrates exceptional performance metrics, making it the algorithm of choice when maximizing accuracy is the primary objective, even at the cost of longer processing times.

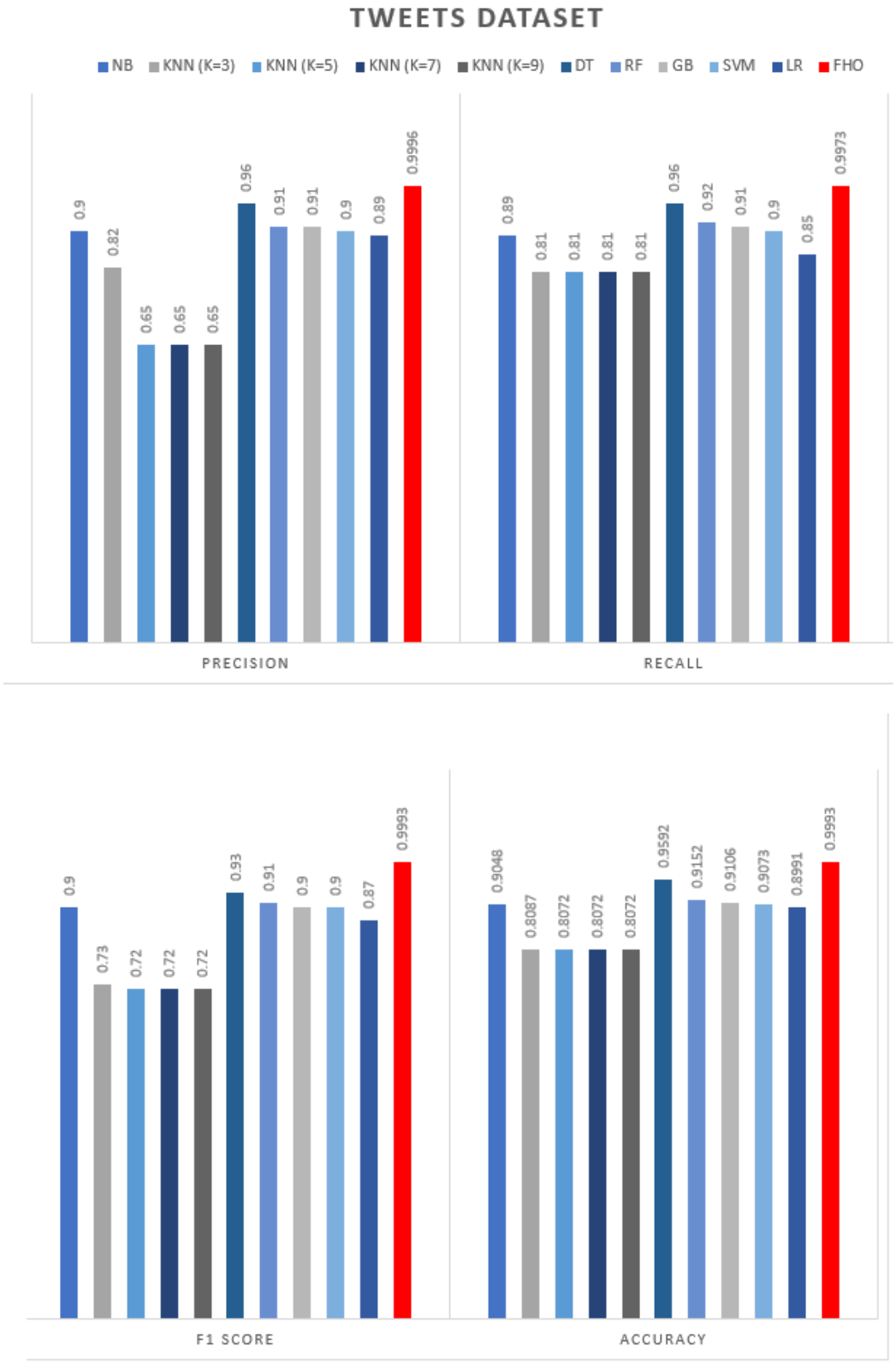

Figure 5.7: Comparison of metrics with FHO on Isis Twitter Dataset

5.3 Conclusion

In conclusion, our experiments have shown that integrating the Firehawk optimizer with supervised techniques yields enhanced results across all four datasets. The application of the Firehawk optimizer improves the performance of supervised methods, resulting in more accurate and effective outcomes during the clustering process. These findings highlight the effectiveness of incorporating the Firehawk optimizer as a robust optimization tool to enhance the performance of supervised techniques in different data clustering scenarios. The successful integration of the Firehawk optimizer demonstrates its potential to optimize the clustering process and achieve improved clustering accuracy. This research provides valuable insights into the benefits of employing the Firehawk optimizer in conjunction with supervised techniques for enhancing clustering performance and achieving superior results in diverse datasets.

Chapter 6

Conclusion

6.1 Summary of our contributions

A new comparative study

On several Online Social Network real-world datasets, we conducted a comparison research of various false profile detection methods in this dissertation. For detecting fake accounts on Facebook, Twitter, and Instagram,even detecting isis fanboys on a special tweeter dataset . we have created and studied supervised and unsupervised algorithms using various performance measures.

It is obvious that none of the techniques had the same results in every situation . Each algorithm performed better in some circumstances while doing worse in others.Furthermore, the results of anomaly detection techniques were found to be influenced by the characteristics of the dataset. While some techniques excelled with smaller datasets, they may not be suitable for larger datasets. Additionally, certain techniques exhibited better accuracies when used on pre-processed or raw unsampled data, as well as feature selection or data normalizing .

In the datasets utilized in this study, Random forest, KNN,Gradient boosting , SVM , naive bayes and Decision tree are the supervised models that are most likely to perform well. and for the unsupervised models most of the times hierarchical clustering is preforming better than kmeans .

Applying only one machine learning technique won't yield the greatest results. Therefore, integrating several strategies may be employed to obtain higher performance for the detection of fake profiles, which is why we have used the fire hawk algorithm to enhance the performance of a chosen supervised model for the goal of detecting spam users. This

project's key accomplishment was that it improved understanding of the algorithms. The algorithms behaved differently when put through a variety of tests, which made it easier for us to comprehend how they functioned.

we have chosen accuracy , precision , recall and f1score as our metrics . FHO is used to compare the output from the other machine learning models, and the best model is selected.The results we obtained show that the fire hawk method outperforms other traditional learning algorithms in terms of performance.

6.2 Future works

This comparative study has the potential for future enhancement by incorporating and contrasting additional bio-inspired models using both supervised and unsupervised methodologies. By expanding the scope of the comparison, researchers can gain a more comprehensive understanding of the strengths and weaknesses of different algorithms. This, in turn, can lead to the identification of the best available solution for a given problem such as Fake profiles detection in online social networks.

To achieve this, future research can consider including a wider range of bio-inspired models, such as genetic algorithms, particle swarm optimization, ant colony optimization, and others. These algorithms can be compared not only in terms of their performance but also their underlying principles and mechanisms.

In addition to algorithmic comparisons, it would be beneficial to explore the applicability of the proposed FHO (Fire Hawk Optimizer) algorithm in practical domains. While the study utilized mathematical test functions on the fake profiles detection problem , future research can investigate how well the FHO algorithm performs in real-world optimization problems. This could involve applying the algorithm to various domains such as engineering, finance, logistics, and healthcare, among others.

Furthermore, it would be valuable to assess the performance of the FHO algorithm in scenarios with bound constraints. Many practical optimization problems involve constraints on variables, and evaluating the algorithm's ability to handle such constraints would enhance its practical applicability.

To thoroughly assess the potential of the FHO algorithm and validate its performance, future research can conduct further evaluations and analyses. This includes assessing computational cost and complexity, comparing it against other state-of-the-art algorithms, and benchmarking it on a wider range of test functions and real-world problems.

By undertaking these future endeavors, Scholars can contribute to the advancement of metaheuristic algorithms and their practical applications in optimization tasks. The insights gained from these studies will not only enhance our understanding of different algorithmic approaches but also enable us to solve complex real-world problems more effectively.

Bibliography

[1] K Ming Leung. Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007:123–156, 2007.

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

[3] Vikrant A Dev and Mario R Eden. Formation lithology classification using scalable gradient boosted decision trees. *Computers & Chemical Engineering*, 128:392–404, 2019.

[4] 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.

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

[6] 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.

[7] Xin-She Yang. Optimization and metaheuristic algorithms in engineering. *Metaheuristics in water, geotechnical and transport engineering*, 1:23, 2013.

[8] Kenneth Sörensen. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.

[9] N Pazhaniraja, P Victer Paul, G Roja, K Shanmugapriya, and Banwait Sonali. A study on recent bio-inspired optimization algorithms. In *2017 Fourth International*

Conference on Signal Processing, Communication and Networking (ICSCN), pages 1–6. IEEE, 2017.

[10] Mahdi Azizi, Siamak Talatahari, and Amir H Gandomi. Fire hawk optimizer: A novel metaheuristic algorithm. *Artificial Intelligence Review*, 56(1):287–363, 2023.

[11] Arnab Mitra, Anirban Kundu, Matangini Chattopadhyay, and Avishek Banerjee. An approach to detect fake profiles in social networks using cellular automata-based pagerank validation model involving energy transfer. *SN Computer Science*, 3(6):423, 2022.

[12] Nuhu Ibrahim and Riza Batista-Navarro. Automatic detection of deaths from social networking sites. In *Information Management and Big Data: 8th Annual International Conference, SIMBig 2021, Virtual Event, December 1–3, 2021, Proceedings*, pages 236–252. Springer, 2022.

[13] A Praveena and S Smys. Effective spam bot detection using glow worm-based generalized regression neural network. In *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2021*, pages 469–487. Springer, 2022.

[14] Mazhar Javed Awan, Muhammad Asad Khan, Zain Khalid Ansari, Awais Yasin, and Hafiz Muhammad Faisal Shehzad. Fake profile recognition using big data analytics in social media platforms. *International Journal of Computer Applications in Technology*, 68(3):215–222, 2022.

[15] Vivek Tanniru and Tathagata Bhattacharya. Online fake logo detection system. 2023.

[16] Liesbeth Allein, Marie-Francine Moens, and Domenico Perrotta. Preventing profiling for ethical fake news detection. *Information Processing & Management*, 60(2):103206, 2023.

[17] Sami Ben Jabeur, Hossein Ballouk, Wissal Ben Arfi, and Jean-Michel Sahut. Artificial intelligence applications in fake review detection: Bibliometric analysis and future avenues for research. *Journal of Business Research*, 158:113631, 2023.

[18] Mekhail Mustak, Joni Salminen, Matti Mäntymäki, Arafat Rahman, and Yogesh K Dwivedi. Deepfakes: Deceptions, mitigations, and opportunities. *Journal of Business Research*, 154:113368, 2023.

[19] Boyeon Jang, Sihyun Jeong, and Chong-kwon Kim. Distance-based customer detection in fake follower markets. *Information Systems*, 81:104–116, 2019.

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

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

[22] Ananya Dey, Hamsashree Reddy, Manjistha Dey, Niharika Sinha, and J Joy. Detection of fake accounts in instagram using machine learning. *Int. J. Comput. Sci. Inf. Technol*, 11(5):83–90, 2019.

[23] Nour El Houda Ben Chaabene, Amel Bouzeghoub, Ramzi Guetari, and Henda Hajjami Ben Ghezala. Applying machine learning models for detecting and predicting militant terrorists behaviour in twitter. In *2021 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 309–314. IEEE, 2021.

[24] Keshav Kaushik, Akashdeep Bhardwaj, Manoj Kumar, Sachin Kumar Gupta, and Abhishek Gupta. A novel machine learning-based framework for detecting fake instagram profiles. *Concurrency and Computation: Practice and Experience*, 34(28):e7349, 2022.

[25] Atika Mbarek, Salma Jamoussi, and Abdelmajid Ben Hamadou. An across online social networks profile building approach: Application to suicidal ideation detection. *Future Generation Computer Systems*, 133:171–183, 2022.

[26] Khubaib Ahmed Qureshi, Rauf Ahmed Shams Malick, Muhammad Sabih, and Hocine Cherifi. Deception detection on social media: A source-based perspective. *Knowledge-Based Systems*, 256:109649, 2022.

[27] Pratheeksha Hegde, Nikitha Saurabh, Preethi Salian, et al. Detection and classification of genuine user profile based on machine learning techniques. In *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pages 1–6. IEEE, 2022.

[28] S Siva Rama Krishna, K Umakanth Reddy, T Anji Reddy, A Saiteja, and R Sumanjali. Detection of fake and clone accounts in twitter using classification and distance measure algorithms. In *Intelligent Manufacturing and Energy Sustainability: Proceedings of ICIMES 2021*, pages 391–399. Springer, 2022.

[29] Faouzia Benabbou, Hanane Boukhouima, and Nawal Sael. Fake accounts detection system based on bidirectional gated recurrent unit neural network. *International Journal of Electrical & Computer Engineering (2088-8708)*, 12(3), 2022.

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

[31] Putra Wanda. Runmax: fake profile classification using novel nonlinear activation in cnn. *Social Network Analysis and Mining*, 12(1):158, 2022.

[32] Mariam Elhussein. Is it sarrah rahamah? a supervised classification model to detect fake identities on facebook within the sudanese community. *Personal and Ubiquitous Computing*, 27(1):107–118, 2023.

[33] N Deshai, B Bhaskara Rao, et al. Deep learning hybrid approaches to detect fake reviews and ratings. *Journal of Scientific & Industrial Research*, 82(1):120–127, 2022.

[34] A Saravanan and Vineetha Venugopal. Detection and verification of cloned profiles in online social networks using mapreduce based clustering and classification. *International Journal of Intelligent Systems and Applications in Engineering*, 11(1):195–207, 2023.

[35] 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.

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

[37] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[38] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Boosting and additive trees. *The elements of statistical learning: data mining, inference, and prediction*, pages 337–387, 2009.

[39] S Madeh Piryonesi and Tamer E El-Diraby. Using machine learning to examine impact of type of performance indicator on flexible pavement deterioration modeling. *Journal of Infrastructure Systems*, 27(2):04021005, 2021.

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

[41] 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.

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

[43] Laura Calvet, Jésica de Armas, David Masip, and Angel A Juan. Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. *Open Mathematics*, 15(1):261–280, 2017.

[44] Jeffrey R Sampson. Adaptation in natural and artificial systems (john h. holland), 1976.

[45] Rainer Storn and Kenneth Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341, 1997.

[46] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308, 2003.

[47] Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8:239–287, 2009.

[48] Stefan Droste, Thomas Jansen, and Ingo Wegener. Optimization with randomized search heuristics—the (a) nfl theorem, realistic scenarios, and difficult functions. *Theoretical Computer Science*, 287(1):131–144, 2002.

[49] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[50] Satyasai Jagannath Nanda and Ganapati Panda. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary computation*, 16:1–18, 2014.

[51] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308, 2003.

[52] D Binu and BS Kariyappa. Ridenn: A new rider optimization algorithm-based neural network for fault diagnosis in analog circuits. *IEEE Transactions on Instrumentation and Measurement*, 68(1):2–26, 2018.

[53] Shinsiong Pang and Mu-Chen Chen. Optimize railway crew scheduling by using modified bacterial foraging algorithm. *Computers & Industrial Engineering*, 180:109218, 2023.

[54] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[55] Marco Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano*, 1992.

[56] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826(1989):37, 1989.

[57] Ashima Yadav and Dinesh Kumar Vishwakarma. A comparative study on bio-inspired algorithms for sentiment analysis. *Cluster Computing*, 23:2969–2989, 2020.

[58] Nour El Houda Ben Chaabene, Amel Bouzeghoub, Ramzi Guetari, and Henda Hajjami Ben Ghezala. Applying machine learning models for detecting and predicting

militant terrorists behaviour in twitter. In *2021 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 309–314. IEEE, 2021.

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