Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 1, January 2025, pp. 257~268
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i1.pp257-268

Enhancing fake profile detection through supervised and hybrid machine learning: a comparative analysis

Ismail Bensassi, Oussama Ndama, Mohamed Kouissi, El Mokhtar En-Naimi
Faculty of Science and Technology of Tangier, Abdelmalek Essaadi University of Tetuan, Tangier, Morocco

Article Info

Article history:

Received Jun 28, 2024
Revised Aug 28, 2024
Accepted Sep 2, 2024

Keywords:

Bernoulli Naïve Bayes
Fake profiles detection
K-medoids
Linear SVC and K-means
Logistic regression
Supervised and unsupervised machine learning algorithms as KNN, SVM
User behavior

ABSTRACT

In modern times, social networks have become ubiquitous platforms facilitating widespread information dissemination, resulting in significant daily data generation. This increase in data production encompasses a wide range of user-generated content, which in turn promotes the proliferation of fraudulent users creating fake profiles and engaging in deceptive activities. This article aims to address this challenge by employing machine learning algorithms to accurately identify fake profiles. The research involves a thorough analysis of various user behaviors, engagement metrics, and content attributes within social platforms. The primary goal is to develop robust models capable of effectively detecting deceptive profiles by meticulously examining user activities and content characteristics. The study explores the application of robust methodologies such as K-means and K-medoids clustering, alongside supervised machine learning classifiers including K-nearest neighbors (KNN), support vector machine (SVM), Bernoulli Naïve Bayes (NB), logistic regression, and linear support vector classification (SVC), specifically tailored for the detection of fake profiles.

Corresponding Author:

Ismail Bensassi
Faculty of Science and Technology of Tangier, Abdelmalek Essaadi University of Tetuan
B.P 416, postal code: 90000, Tangier, Morocco
Email: bensassi.ismail@gmail.com

1. INTRODUCTION

The rapid expansion of social media platforms in recent years has profoundly impacted various societal realms, particularly in marketing and celebrity self-promotion, which have significantly broadened their follower and fan bases. While these platforms offer extensive benefits such as widespread information dissemination, they also introduce challenges related to privacy, misinformation, and online security. A critical issue among these challenges is the widespread presence of fake profiles, which pose risks by disseminating misinformation, engaging in fraudulent activities, and compromising the privacy of genuine users. Addressing the complex challenges posed by fake profiles requires a comprehensive approach integrating technological solutions aimed at detecting, preventing, and mitigating the negative impacts associated with these deceptive entities. This study focuses on harnessing machine learning algorithms to accurately detect fake profiles on social networks. By meticulously analyzing user behaviors, engagement metrics, and content characteristics within these platforms, the goal is to develop robust models capable of effectively identifying deceptive profiles.

Existing studies are carried out with supervised algorithms which are generally more accurate for specific tasks with labeled data, while unsupervised algorithms are better for discovering hidden structures in unlabeled data. In our approach, we have opted for a mixture of both types of algorithm, which allows us to take advantage of the strengths of each method, using unsupervised algorithms to explore and prepare the data before applying supervised methods for accurate predictions. This combination is often used to optimize performance in situations where labeled data is scarce or expensive to obtain, and also to get the best results in terms of time and accuracy in detecting false profiles in a big data context.

This paper's primary contribution lies in developing a model that utilizes various machine learning algorithms to analyze large datasets and identify patterns indicative of fake profiles with heightened accuracy. Additionally, the study identifies and examines pertinent features that distinguish genuine from fake profiles, such as user followers, friends, favorites, and status count, providing valuable insights into the characteristics of deceptive accounts. The remainder of this paper is structured as follows: section 1 provides an extensive literature review focusing on different machine learning techniques employed in detecting fake profiles. Section 2 elaborates on the methodology used in this study, detailing the dataset, data preprocessing steps, and the extracted features for analysis. In section 3 presents the results and discusses findings, emphasizing the effectiveness of the proposed approach. Finally, section 4 draws conclusions from the study and outlines recommendations for future research efforts in this field.

2. STUDIES AND METHOD

2.1. Related studies

The literature review comprehensively examines previous studies and methodologies employed for detecting fake profiles on social media platforms. Various methods categorize profiles based on account activity, response rates to requests, volume of communications, and other profile characteristics. Some models are graph-based, while others focus on distinguishing between cyborgs and bots using specific techniques. Additionally, spam detection based on certain keywords in messages has been applied to identify fraudulent social media profiles. In 2008, Sybil Rank was developed to mitigate the impact of Sybil attacks on social media, restricting random walk encounters and utilizing Kleinberg's synthetic social network dataset. Concurrently, Sybil Restriction employed a similar approach but emphasized rapid convergence outside the Sybil zone using random variables and frequency-based ranking. In 2009, Sybil-infer utilized randomized walks and rapid convergence outside the non-Sybil area, employing model-based sampling, greedy algorithms, and Bayesian networks with probabilistic threshold selection. Mislove's algorithm in 2010 selected profiles from the Facebook dataset based on metric-adjusted conductivity using greedy search. The Facebook Immune System, introduced in 2011, integrated random forest, support vector machine (SVM), and boosting techniques for detection, employing feature loops selection on the Facebook dataset. Ramalingam and Chinnaiah [1], Hajdu *et al.* [2] introduced a regression approach ranking profiles based on interactions, tagging, and wall postings, where lower rankings indicate false accounts compared to genuine ones, although its reliability was questioned. Swe and Myo [3] developed a blacklist distinguishing between fake features and accounts, while another proposed Wanda and Jie [4], utilizing a supervised learning algorithm within a dynamic CNN framework for detection. Kodati *et al.* [5] combined SVM, random forest, and AdaBoost for detecting fake online social network (OSN) accounts, Meshram *et al.* [6] specifically used regression analysis and random forest classifiers to identify fake Instagram accounts. Various related works by Chakraborty *et al.* [7] and Ahmed *et al.* [8] have also contributed to this evolving field.

2.2. Method (machine learning classification and clustering models)

2.2.1. Supervised machine learning algorithms

A. The K-nearest neighbors machine learning algorithm:

The K-nearest neighbors (KNN) algorithm is a straightforward method for data classification. It operates as a non-parametric, supervised machine learning algorithm suitable for both classification and regression tasks. The fundamental goal of KNN is to classify new data points based on the attributes of nearby training data points [9]. Here's how the algorithm typically works:

- Specify the number of nearest neighbors KKK to consider.
- Define the training dataset D(yj) D(y_j) D(yj).
- Calculate the distance d(xi, yj) d(x_i, y_j) d(xi, yj) between the new observation xix_ixi and each point yjy_jyj in the training data. Various distance metrics can be used, with Euclidean distance being the default choice.
- Sort the distances in ascending order.
- Select the first KKK entries with the smallest distances.
- Classify the observation xix_ixi based on the majority class of its KKK nearest neighbors.

KNN offers several advantages:

- Simplicity and ease of implementation: KNN is straightforward to understand and implement, making it accessible even for beginners in machine learning.

– Versatility: it can be applied to classification, regression, and search problems. In classification tasks, it predicts the class based on the most prevalent class among its neighbors. For regression, it predicts the value based on the average (or weighted average) of its neighbors' values.

Overall, KNN is valued for its simplicity, versatility across different types of problems, and effectiveness in scenarios where the decision boundary is not linear or where the data distribution is not known in advance.

B. The support vector machine machine learning algorithm

The SVM is a supervised machine learning algorithm utilized for classification, regression, and anomaly detection [10]. It originated in 1992 from a statistical learning theory developed by Vladimir Vapnik, Boser, and Guyon, gaining rapid adoption due to its capability to handle high-dimensional data, theoretical underpinnings, and practical effectiveness. SVMs are appreciated for their simplicity, requiring few parameters compared to other models. In practice, SVMs are widely used for binary classification tasks [11], implemented through libraries such as Scikit-learn in Python (sklearn.svm module). This allows users to solve both classification and regression problems efficiently.

The SVM operates by finding optimal decision boundaries or hyperplanes as shown in in an N-dimensional space [12], where N represents the number of features in the dataset. These hyperplanes act as decision boundaries to distinctly classify data points [13]. The primary objective of SVM is to maximize the margin between classes, which refers to the distance between the nearest data points of different classes. Instances of data that lie closest to these separating hyperplanes are termed support vectors [14], [15]. Maximizing this margin allows for more confident classification of future data points, can be seen in Figure 1.

The versatility of SVM extends to various applications, including:

– Image classification: SVMs are known for achieving high accuracy in tasks such as image classification and segmentation.
– Text and hypertext categorization: they can reduce the dependency on labeled data, making them effective in categorizing textual and web content.
– Handwritten characters recognition: SVMs exhibit strong performance in recognizing handwritten characters.
– Proteins classification in biology: they are applied to classify proteins in biological research.

Overall, SVMs are robust tools in machine learning, valued for their ability to handle complex datasets and deliver reliable results across diverse domains.

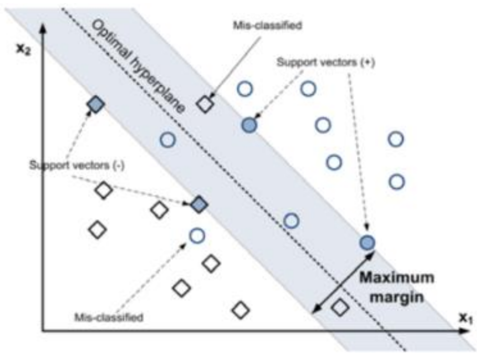


Figure 1. Optimal hyper plane and support vectors

C. The logistic regression

Logistic regression is a statistical method primarily employed for binary classification tasks, predicting the probability that an instance belongs to a specific category. Despite its name, logistic regression functions as a classification algorithm rather than a regression algorithm. It is designed to model the probability of an instance belonging to a particular class based on input features.

This method is particularly well-suited for binary classification problems where the outcome variable (dependent variable) has two possible outcomes. Logistic regression is chosen when the dependent variable is dichotomous, meaning it has two categories or levels. It is used to analyze and explain the relationship between a binary outcome variable and one or more independent variables, which can be nominal, ordinal, interval, or ratio-level variables [16].

In essence, logistic regression estimates the probability of a binary outcome based on predictor variables, providing insights into the likelihood of an instance belonging to a specific class. It is widely applied in various fields for tasks such as predicting customer churn, medical diagnoses, credit scoring, and more, where understanding the probability of an event occurring is crucial for decision-making.

D. The linear support vector classification

Linear support vector classification (SVC), short for linear SVC, is a variant of the SVM algorithm tailored for both binary and multiclass classification tasks. Unlike traditional SVMs that seek a hyperplane to separate data into distinct classes, linear SVC specifically focuses on achieving linear separation. Its objective is to identify a hyperplane within the feature space that effectively divides instances belonging to different classes.

The primary aim of linear SVC is to establish a hyperplane that maximizes the margin of separation between classes. This margin represents the distance between the hyperplane and the closest data points from each class, known as support vectors. By optimizing a convex quadratic programming problem, linear SVC determines the coefficients and bias terms that define the separating hyperplane. This optimization process aims to minimize classification errors while ensuring the largest possible margin between classes.

In practical terms, linear SVC is advantageous for scenarios where the relationship between features and classes can be effectively modeled using a linear decision boundary. It is widely used in applications such as text categorization, image recognition, and sentiment analysis, where linear separation suffices to achieve accurate classification results. Overall, linear SVC provides a computationally efficient approach to binary and multiclass classification problems by leveraging the principles of SVMs, emphasizing simplicity and effectiveness in separating data points across different classes.

E. The Bernoulli Naive Bayes (Bernoulli NB)

The Bernoulli NB algorithm is a probabilistic classification method belonging to the NB family. It is specifically designed for binary classification tasks where each feature represents a binary attribute. At its core, the NB algorithm operates based on Bayes' theorem [17], which calculates the probability of a hypothesis given observed evidence. The "Naïve" assumption in NB is that features are conditionally independent given the class label, simplifying probability calculations.

Bernoulli NB is a specialized variant of NB suited for datasets with binary features, where each feature is a binary variable indicating presence or absence (e.g., presence of a term in a document, spam or not spam). In Bernoulli NB, the input is a feature vector where each element represents a binary variable. Typically, the presence of a feature is denoted by 1, and its absence by 0.

In practice, Bernoulli NB is effective for tasks such as text classification (e.g., identifying spam emails), where features are binary indicators of specific terms or characteristics. Its simplicity and efficiency make it suitable for large-scale datasets and real-time applications, leveraging the probabilistic framework of NB to make reliable predictions based on binary feature inputs.

2.2.2. Unsupervised machine learning algorithms

A. The K-means algorithm

Data clustering involves grouping items so that those within the same group are more similar to each other than to those in other groups [18]. The primary goal is to identify homogeneous subgroups within the data that share common characteristics [19], using similarity measures like Euclidean distance or correlation-based metrics. Unlike supervised learning, clustering is unsupervised, meaning there are no predefined labels to guide the grouping process; instead, it aims to uncover inherent structures within the data.

Among clustering algorithms, K-means is widely recognized and utilized. It operates iteratively to partition a dataset into K-distinct clusters, where each data point belongs exclusively to one cluster [20]. The fundamental objective of K-means is to minimize intra-cluster variability while maximizing inter-cluster differences [21]. Essentially, this means making data points within each cluster as similar as possible while ensuring dissimilarity between clusters.

The K-means algorithm follows these steps in unsupervised machine learning:

- Specify the number of clusters KKK.
- Initialize centroids: initially, shuffle the dataset and randomly select K-data points as centroids without replacement.
- Iterate until convergence: continue iterating until centroids no longer change, indicating stabilization in cluster assignments.
- Assign data points to clusters: calculate the distance between each data point and all centroids, assigning each data point to the closest centroid.

– Update centroids: recompute centroids by averaging the positions of all data points assigned to each cluster.

K-means is favored for its ability to handle large datasets efficiently, offering relatively fast computation times [22]. It is commonly applied in various fields for tasks such as customer segmentation, image compression, and anomaly detection, where uncovering natural groupings in data is essential for analysis and decision-making.

B. The K-medoids algorithm

The partitioning around medoids (K-medoids) algorithm is a variant of the K-means algorithm that differs in its use of medoids rather than means as centroids. A medoid within a cluster is an object that minimizes the average dissimilarity to all other objects in that cluster. The K-medoids algorithm starts by computing K-medoids and then assigns each object in the dataset to the nearest medoid based on a chosen distance metric. Subsequently, it evaluates the cost of swapping a data point PiP_iPi with a medoid MiM_iMi to potentially improve the clustering arrangement [23].

To outline the steps of the K-medoids algorithm:

– Step 1: select the initial K points as medoids.
– Step 2: evaluate the cost associated with swapping each selected medoid with each unselected data point.
– Step 3: choose the swap that results in the lowest cost.
– Step 4: repeat until no further improvements in cost can be achieved, where each unselected point is compared to its closest selected medoid [24].

The K-medoids algorithm is effective in scenarios where the choice of a representative object (medoid) is crucial, such as in biological clustering or market segmentation, where identifying the most typical or representative member of a group is essential for accurate analysis and decision-making.

3. RESULTS AND DISCUSSION

3.1. Supervised algorithm application results

3.1.1. Building the classification model

Training: this consists of two modules, training and testing, each using a part of the feature base subdivided into two parts, the training base and the test base. The training module uses the training base to provide a decision model, while the test module uses the test base to measure the performance of the model provided.

Use: this is the last and most important phase in our system. After having a better performance rate, or after having built the best model in the previous phase, we must use it on new unprocessed information, and the model that allows us to predict the class of the new situation if it is false or true with a degree of confidence as shown in Figure 2.

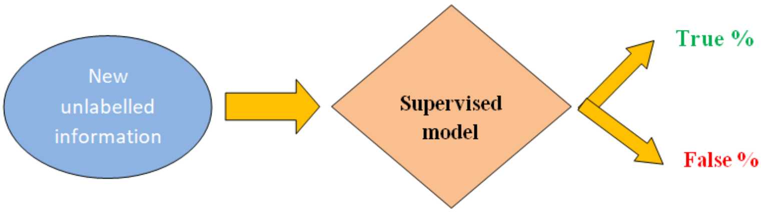


Figure 2. Usage process

In this phase, we apply five supervised machine learning algorithms to the profiles mentioned earlier: logistic regression, linear SVC, KNN, SVM, and Bernoulli NB. These experiments are conducted using Scikit-learn [25], [26], a Python-based machine learning library that offers a wide array of tools and models for streamlined implementation of machine learning algorithms.

Following data cleaning to remove outliers and enhance data quality, we carefully selected a subset of features deemed pertinent for the model development process. These selected features are essential for training and evaluating the performance of each algorithm in our analysis. The Table 1 is an extract from our dataset.

Table 1. Extract from our database

	id	statuses _count	followers_count	freinds_count	favourites _count	listed_count	profile_use_ background_image	isFake
0	3610511	20370	5470	2385	145	52	1.0	0
1	5656162	3131	506	381	9	40	1 .0	0
1	5682702	4024	264	87	323	16	1 .0	0
3	6067292	40586	640	622	1118	32	1.0	0
4	6015122	2016	62	64	13	0	1 .0	0

The purpose of model evaluation is to assess how well a model will perform on new, unseen data, known as out-of-sample data. It is crucial not to evaluate a model on the same data used for training because this can lead to overfitting [27], where the model memorizes the training data rather than learning general patterns, thus hindering its ability to generalize. After training on the training set, the performance of each model is evaluated using the test set, focusing on three key metrics: accuracy, precision, and recall.

– Logistic regression: the logistic regression algorithm achieved an accuracy of 48%, indicating moderate performance.
– Linear SVC: the linear SVC algorithm also achieved an accuracy of 48%, suggesting similar performance to logistic regression.
– SVM: the SVM algorithm achieved a high accuracy of 94%, indicating strong predictive capability.
– KNN: the KNN algorithm achieved a very high accuracy of 97%, demonstrating excellent performance.
– Bernoulli NB: the Bernoulli NB algorithm achieved a high accuracy of 96%, indicating robust performance in classification tasks.

These accuracy scores reflect how well each algorithm generalized to the test data after being trained on the training set. Higher accuracy scores generally indicate better performance, with SVM, KNN, and Bernoulli NB showing particularly strong results in this evaluation. These metrics provide insights into the effectiveness of each algorithm in making accurate predictions on new, unseen data.

3.1.2. Summary table

Table 2 summarizes the prediction accuracy of the algorithms employed in the study. Among them, KNN, SVM, and Bernoulli NB demonstrate notably superior performance compared to linear SVC, SVM, and logistic regression. Specifically, KNN achieves the highest accuracy and precision, achieving approximately 97%. This indicates that KNN effectively predicts outcomes with a high degree of accuracy compared to the other algorithms tested. Overall, the Table 2 underscores the effectiveness of KNN, SVM, and Bernoulli NB in achieving robust prediction accuracy, highlighting their suitability for the specific classification tasks evaluated in this study.

Table 2. Prediction accuracy of our algorithms

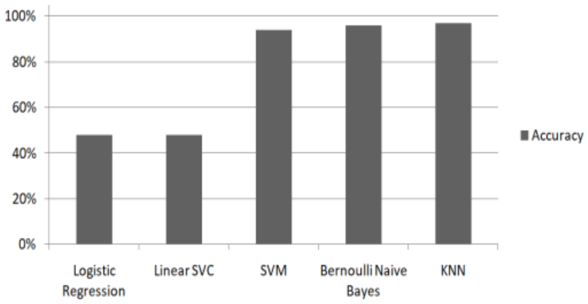
Classification algorithm	Accuracy	Precision
Logistic regression	48%	52%
Linear SVC	48%	52%
SVM	94%	90%
Bernoulli NB	96%	93%
KNN	97%	97%

In summary, our study evaluates several machine learning algorithms using the Scikit-learn library in Python. In Figure 3 results highlight varying levels of accuracy and precision among these algorithms, as depicted in Figures 3(a) and 3(b) which is made up of two graphs, the grey graph shows the accuracy and the blue graph shows the precision resulting from the application of different machine learning algorithms to our test base. Specifically, KNN, Bernoulli NB, and SVM demonstrate significantly higher accuracy rates, achieving approximately 97%, 96%, and 94% respectively. In contrast, logistic regression and linear SVC exhibit comparatively lower accuracy in our evaluation. These findings underscore the effectiveness of KNN, Bernoulli NB, and SVM in accurately predicting outcomes for the tasks considered in our study. They emphasize the importance of selecting appropriate machine learning algorithms based on the specific characteristics and requirements of the dataset and the problem domain.

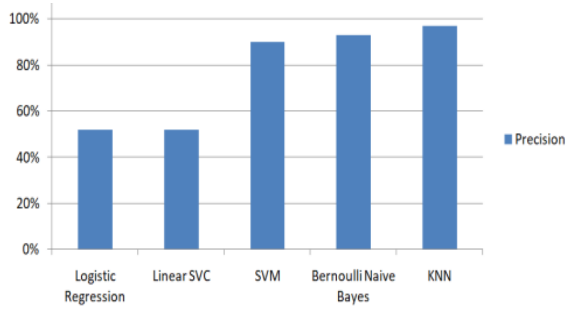
3.2. Effect of unsupervised algorithms

Initially, we remove the segmentation of our dataset by excluding the “is_fake” field. To achieve the highest accuracy for our model, it’s essential to determine the optimal number of clusters, denoted as k, for our dataset. The Elbow method [28]-[30] is widely used to identify this optimal k value. After training our model with various values of k using both the training and testing datasets, we determine that the optimal k value for our model is 3, as illustrated in Figure 4. This approach helps us establish the number of clusters that best organizes our data while maximizing the accuracy of our clustering model.

Figure 5 displays the clustering outcome of our data, where cases are grouped into three distinct clusters represented by colors: yellow, purple, and green. These clusters are determined based on the six parameters depicted in Table 1. This visualization is generated using the matplotlib library in Python [31], [32], allowing us to visually understand how our data points are distributed across different clusters. It provides a clear representation of how the clustering algorithm has organized our dataset based on the selected parameters.



(a)



(b)

Figure 3. The overall prediction (a) accuracy and (b) precision of supervised algorithms

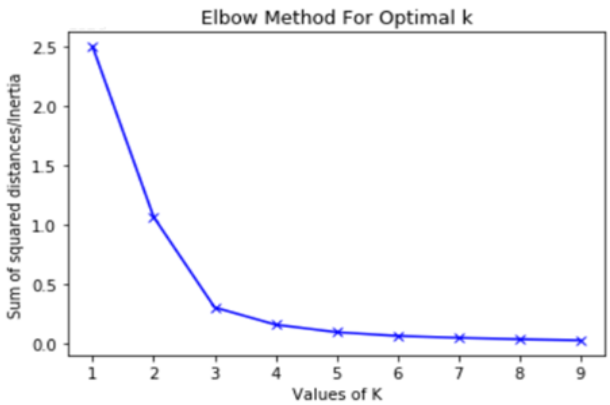


Figure 4. Error rate vs K-value

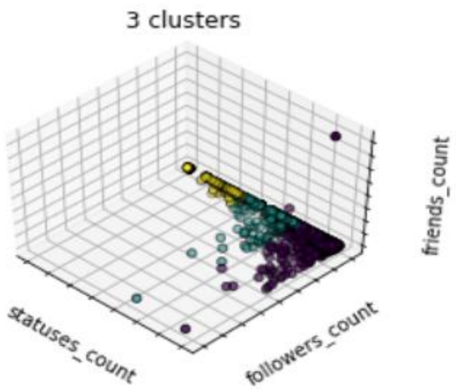


Figure 5. Partition of cases into three clusters

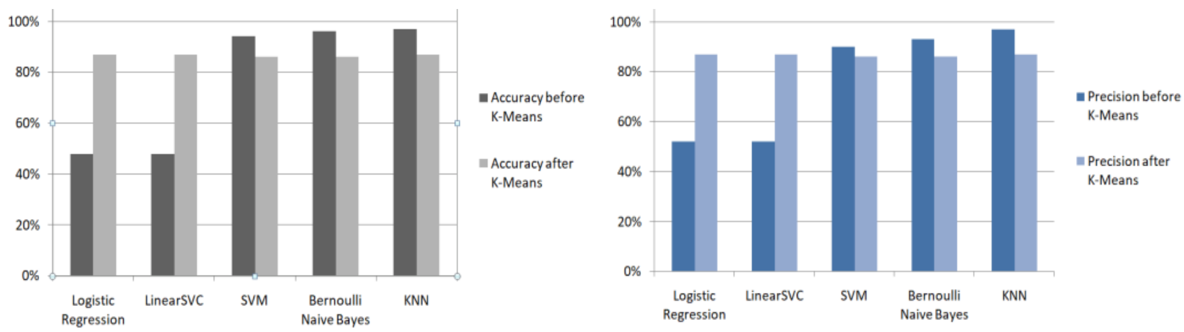
3.2.1. Combination of the K-means and the supervised machine learning algorithms

The Table 3 provides a summary of the impact of the K-means algorithm on the performance of various algorithms previously evaluated (KNN, SVM, linear SVC, Bernoulli NB, logistic regression). Notably, KNN, Bernoulli NB, and SVM demonstrate significantly higher accuracy rates and precision levels (97%, 93%, and 90%, and 87%, 86%, 86% respectively) compared to logistic regression and linear SVC.

Table 3. Performance and precision comparison before and after inclusion of K-means clustering

Classification algorithm	Accuray before K-means	Accuracy after K-means	Precision before K-means	Precision after K-means
Logistic regression	48%	87%	52%	87%
Linear SVC	48%	87%	52%	87%
SVM	94%	86%	90%	86%
Bernoulli NB	96%	86%	93%	86%
KNN	97%	87%	97%	87%

These results underscore the effectiveness of K-means clustering in enhancing the performance of classification algorithms, particularly for tasks where grouping data into clusters based on similarity improves predictive accuracy and precision. They highlight the importance of integrating clustering techniques like K-means to optimize the outcomes of machine learning models in various application domains as indicated in Figure 6 which is made up of two graphs, in Figure 6(a) the graph in gray schematizes the accuracy after applying both the K means algorithm of clustering with the various supervised machine learning algorithms above-mentioned, to our test base and the graph in blue schematizes the precision in Figure 6(b).



(a) (b)

Figure 6. The overall prediction (a) accuracy and (b) precision of supervised algorithms

The impact of the K-means algorithm on our data is striking, particularly in enhancing both precision and accuracy across several algorithms that initially yielded lower results. This improvement is especially notable for algorithms that previously showed lower performance metrics. K-means clustering has effectively contributed to refining the predictions and classifications made by these algorithms, highlighting its role in optimizing model outcomes through better data organization and cluster-based analysis.

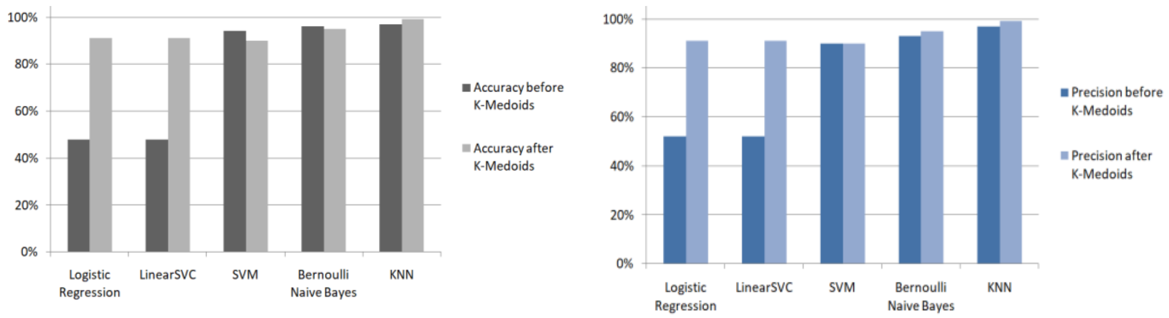
3.2.2. Combination of the K-medoids and the supervised machine learning algorithms

Table 4 provides a summary of how the K-medoids algorithm influences the performance of the algorithms analyzed in the previous section (KNN, SVM, linear SVC, Bernoulli NB, and logistic regression). Indeed, KNN, Bernoulli NB, and SVM demonstrate significantly higher accuracy and precision rates (97%, 93%, and 90%, and 99%, 95%, and 90% respectively) compared to logistic regression and linear SVC.

Table 4. Performance and precision comparison before and after inclusion of K-medoids clustering

Classification algorithm	Accuray before K-medoids	Accuracy after K-medoids	Precision before K-medoids	Precision after K-medoids
Logistic regression	48%	91%	52%	91%
Linear SVC	48%	91%	52%	91%
SVM	94%	90%	90%	90%
Bernoulli NB	96%	95%	93%	95%
KNN	97%	99%	97%	99%

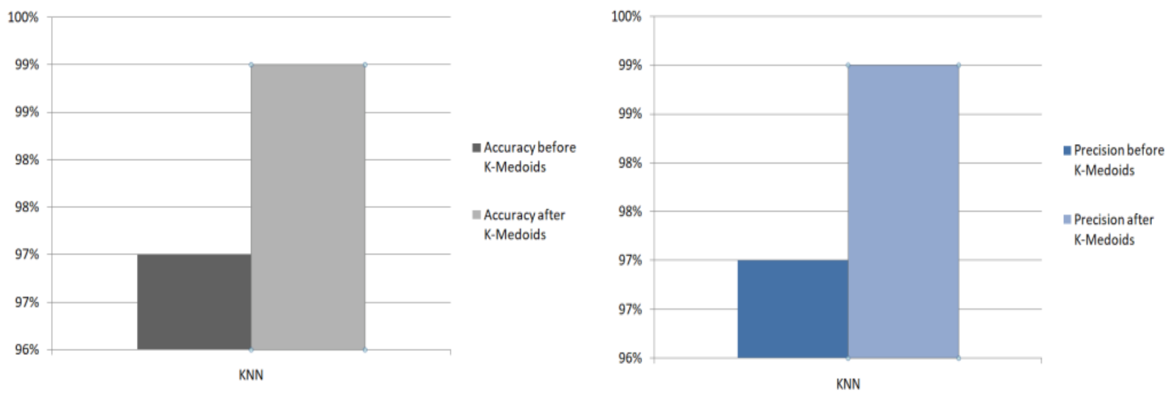
These results underscore the impactful role of the K-medoids algorithm in enhancing the predictive capabilities of these algorithms. By focusing on medoids rather than centroids, K-medoids effectively refines clustering decisions, thereby optimizing the performance of classification algorithms in various data-driven tasks which explains the values shown in Figure 7. Which is made up of two graphs, in the Figure 7(a) gray schematizes the accuracy after applying both the K-medoids as unsupervised algorithms with the various supervised machine learning algorithms above-mentioned, to our test base and the graph in blue schematizes the precision in Figure 7(b).



(a) (b)

Figure 7. The performance of (a) accuracy and (b) precision before and after K-medoids algorithm

These findings indicate that incorporating clustering methods, particularly K-means and K-medoids, positively influences the predictive capability of classification algorithms for identifying fake profiles. The substantial enhancements in accuracy and precision underscore the effectiveness of clustering techniques in bolstering the overall performance of machine learning models. Figure 8 illustrates the superior performance of this hybrid approach, highlighting its efficacy in accurately classifying and identifying fake profiles within social networks where the Figure 8(a) grey shows the resulting upper accuracy by applying the K-medoids algorithm with the KNN as supervised machine learning algorithm, and the graph in blue shows in Figure 8(b) the precision of the latter.



(a) (b)

Figure 8. The KNN algorithm (a) accuracy and (b) precision before and after K-medoids algorithm

In our specific dataset, the optimal results in terms of accuracy and precision are achieved through a hybrid approach: using K-medoids as an unsupervised algorithm to partition the dataset into cohesive clusters, and then employing KNN as a supervised algorithm for detecting fake profiles. This is why our approach is so important compared to existing approaches in terms of accuracy, with a significant jump in the value of the latter, approaching 100% with a value of 99%. As a result, the objective of marrying supervised and unsupervised algorithms is achieved, and our studies guarantee good results in predicting fake profiles.

4. CONCLUSION AND PERSPECTIVES

The study presented a comprehensive methodology aimed at detecting fake profiles on social media platforms by leveraging multiple supervised machine learning algorithms. The research involved an extensive exploration of various user behaviors, engagement metrics, and content attributes to construct robust models capable of accurately identifying deceptive profiles. Initially, the dataset underwent clustering using unsupervised machine learning algorithms such as K-means initially and subsequently K-medoids, grouping similar profiles based on shared characteristics. This clustering process facilitated the identification of cohesive groups within the dataset, enhancing the efficiency of subsequent classification tasks.

Following clustering, classification tasks were performed using a range of supervised machine learning algorithms including KNN, SVM, Bernoulli NB, logistic regression, and linear SVC. These algorithms utilized the similarities in profile features to effectively classify profiles as either genuine or fake.

The results of the study demonstrated the efficacy of the proposed methodology in detecting fake profiles, achieving promising outcomes in terms of accuracy and reliability. By combining clustering and classification techniques, the model effectively distinguished deceptive profiles from genuine ones, offering valuable insights for combating fraudulent activities on social media platforms.

Looking forward, future research will focus on further validating the model using larger and more diverse datasets collected from various social media platforms. Additionally, alternative machine learning and deep learning approaches will be explored to enhance detection accuracy and robustness. This includes investigating ensemble learning methods, neural network architectures, and advanced feature engineering techniques. Moreover, future studies will aim to develop adaptive models capable of dynamically adjusting to evolving patterns of fake profile creation and detection. By integrating real-time data analysis and continuous model refinement, the objective is to create more resilient detection systems that can effectively mitigate the spread of fake profiles on social media platforms.

In conclusion, the study establishes a foundation for ongoing research in the field of fake profile detection, illustrating the potential of machine learning algorithms to address this persistent challenge. Through continuous advancements and research efforts, the goal is to develop sophisticated detection systems that uphold the integrity and trustworthiness of social media platforms in the face of evolving threats posed by deceptive entities. These systems could also be applied across various domains, including social networks, e-commerce platforms, forums, and banking web sites. In the future, we hope to use other machine and deep learning techniques and apply our approach in other relevant areas that affect people's lives, in order to minimize attacks, scams and fraud in big data systems.

REFERENCES

[1] D. Ramalingam and V. Chinnaiah, "Fake profile detection techniques in large-scale online social networks: a comprehensive review," *Computers and Electrical Engineering*, vol. 65, pp. 165–177, Jan. 2018, doi: 10.1016/j.compeleceng.2017.05.020.

[2] G. Hajdu, Y. Minoso, R. Lopez, M. Acosta, and A. Elleithy, "Use of artificial neural networks to identify fake profiles," in *2019 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2019*, May 2019, pp. 1–4, doi: 10.1109/LISAT.2019.8817330.

[3] M. M. Swe and N. N. Myo, "Fake accounts detection on twitter using blacklist," in *Proceedings - 17th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2018*, Jun. 2018, pp. 562–566, doi: 10.1109/ICIS.2018.8466499.

[4] P. Wanda and H. J. Jie, "DeepProfile: Finding fake profile in online social network using dynamic CNN," *Journal of Information Security and Applications*, vol. 52, p. 102465, Jun. 2020, doi: 10.1016/j.jisa.2020.102465.

[5] S. Kodati, K. P. Reddy, S. Mekala, P. L. S. Murthy, and P. C. S. Reddy, "Detection of fake profiles on twitter using hybrid SVM algorithm," *E3S Web of Conferences*, vol. 309, p. 01046, Oct. 2021, doi: 10.1051/e3sconf/202130901046.

[6] E. P. Meshram, R. Bhambulkar, P. Pokale, K. Kharbikar, and A. Awachat, "Automatic detection of fake profile using machine learning on Instagram," *International Journal of Scientific Research in Science and Technology*, pp. 117–127, May 2021, doi: 10.32628/ijsrst218330.

[7] P. Chakraborty, C. S. Muzammel, M. Khatun, S. F. Islam, and S. Rahman, "Automatic student attendance system using face recognition," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 3, pp. 93–99, Feb. 2020, doi: 10.35940/ijeat.b4207.029320.

[8] M. Ahmed, P. Chakraborty, and T. Choudhury, "Bangla document categorization using deep RNN model with attention mechanism," in *Lecture Notes in Networks and Systems*, vol. 291, 2022, pp. 137–147.

[9] Z. Rustam, S. Hartini, I. Wirasati, and J. E. Aurelia, "An approach for COVID-19 detection using deep convolutional features on chest X-ray images," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 6, pp. 1452–1460, 2021.

[10] A. Ouhmida, A. Raihani, B. Cherradi, and O. Terrada, "A novel approach for parkinson's disease detection based on voice classification and features selection techniques," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 17, no. 10, pp. 111–130, Oct. 2021, doi: 10.3991/ijoe.v17i10.24499.

[11] L. Wang *et al.*, "An intelligent gastric cancer screening method based on convolutional neural network and support vector machine," *International Journal of Computers and Applications*, vol. 43, no. 7, pp. 720–725, 2021, doi: 10.1080/1206212X.2019.1640345.

[12] Z. A. Kakarash, H. S. Ezat, S. A. Omar, and N. F. Ahmed, "Time series forecasting based on support vector machine using particle swarm optimization," *International Journal of Computing*, vol. 21, no. 1, pp. 76–88, Mar. 2022, doi: 10.47839/ijc.21.1.2520.

[13] E. M. Cherrat, R. Alaoui, and H. Bouzahir, “Score fusion of finger vein and face for human recognition based on convolutional neural network model,” *International Journal of Computing*, vol. 19, no. 1, pp. 11–19, Mar. 2020, doi: 10.47839/ijc.19.1.1688.
[14] D. Conforti, D. Costanzo, and R. Guido, “Cancer prognostic evaluation via support vector machines,” *International Journal of Computing*, pp. 29–34, Aug. 2014, doi: 10.47839/ijc.3.3.302.
[15] “towardsdatascience,” [Online]. Available: https://towardsdatascience.com.
[16] D. Novaliendry *et al.*, “Hemodialysis patient death prediction using logistic regression,” *International journal of online and biomedical engineering*, vol. 19, no. 9, pp. 66–80, Jul. 2023, doi: 10.3991/ijoe.v19i09.40917.
[17] I. Marzuki, H. Norlida, and S. B. Salem, “Comparative analysis of Naive Bayesian techniques in health-related for classification task,” *Journal of Soft Computing and Data Mining*, vol. 1, no. 2, 2020.
[18] L. Buitinck *et al.*, “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, [Online]. Available: http://scikit-learn.sourceforge.net.
[19] M. Kalra, N. Lal, and S. Qamar, “K-mean clustering algorithm approach for data mining of heterogeneous data,” in *Lecture Notes in Networks and Systems*, vol. 10, 2018, pp. 61–70.
[20] K. D. Joshi and P. S. Nalwade, “Modified K-means for better initial cluster centres,” *International Journal of Computer Science and Mobile Computing* , vol. 2, no. 7, pp. 219–223, 2013, [Online]. Available: www.ijcsmc.com.
[21] B. K. Khotimah, F. Irhamni, and T. Sundarwati, “A genetic algorithm for optimized initial centers K-means clustering in SMEs,” *Journal of Theoretical and Applied Information Technology*, vol. 90, no. 1, pp. 23–30, 2016.
[22] M. Kouissi, E. M. En-Naimi, and A. Zouhair, “Hybrid approach for wind turbines power curve modeling founded on multi-agent system and two machine learning algorithms, K-means method and the K-nearest neighbors, in the retrieve phase of the dynamic case based reasoning,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 18, no. 06, pp. 110–122, May 2022, doi: 10.3991/ijoe.v18i06.29565.
[23] S. A. Abbas, A. Aslam, A. U. Rehman, W. A. Abbasi, S. Arif, and S. Z. H. Kazmi, “K-means and K-medoids: cluster analysis on birth data collected in city Muzaffarabad, Kashmir,” *IEEE Access*, vol. 8, pp. 151847–151855, 2020, doi: 10.1109/ACCESS.2020.3014021.
[24] V. K. Thambusamy and K. Saravananathan, “Quality based analysis of clustering algorithms using diabetes data for the prediction of disease,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11S2, pp. 448–452, Oct. 2019, doi: 10.35940/ijitee.k1072.09811s219.
[25] F.-Z. Hibbi, O. Abdoun, and E. K. Haimoudi, “Smart tutoring system: a predictive personalized feedback in a pedagogical sequence,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 20, p. 263, Oct. 2021, doi: 10.3991/ijet.v16i20.24783.
[26] L. Buitinck *et al.*, “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, 2011. http://scikit-learn.sourceforge.net. (accessed Aug. 10, 2019).
[27] A Dictionary of Epidemiology (DRAFT). 2014. doi: 10.1093/acref/9780199976720.001.0001.
[28] R. Ahuja, A. Solanki, and A. Nayyar, “Movie recommender system using K-means clustering and K-nearest neighbor,” *Conference: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2019, doi: 10.1109/confluence.2019.8776969.
[29] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, “Integration K-means clustering method and elbow method for identification of the best customer profile cluster,” *IOP Conference Series: Materials Science and Engineering*, vol. 336, no. 1, p. 012017, Apr. 2018, doi: 10.1088/1757-899X/336/1/012017.
[30] S. Rakic, N. Tasic, U. Marjanovic, S. Softic, E. Lüftenegger, and I. Turcin, “Student performance on an e-learning platform: mixed method approach,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 02, p. 187, Jan. 2020, doi: 10.3991/ijet.v15i02.11646.
[31] “Matplotlib — Visualization with Python.” https://matplotlib.org/ (accessed Aug. 17, 2024).
[32] “Welcome to Python.org,” Python.org. https://www.python.org/ (accessed Aug. 17, 2024).

BIOGRAPHIES OF AUTHORS

Ismail Bensassi is a Ph.D. student in E-DSAI2S (Research team in Data Science, Artificial Intelligence, and Smart Systems), C3S Computer Science and Smart Systems Laboratory, Faculty of Sciences and Technologies (FST), Tangier, Morocco. He is an Engineer in computer Science, Laureate of FST of Tangier, he is a computer systems analyst engineer at the Ministry of Justice, Department of Modernization and Computer Systems, Rabat, Morocco. The research topics of interest are smart systems, sentiment analysis, smart cities, multi-agent systems (MAS), case-based reasoning (CBR), ontology, machine learning, and deep learning. He can be contacted at email: bensassi.ismail@gmail.com.

Oussama Ndama is a Ph.D. student in DSAI2S (Data Science, Artificial Intelligence, and Smart Systems Research Team), C3S Computer Science and Smart Systems Laboratory, Faculty of Sciences and Technologies (FST), Tangier, Morocco. He had his Master in Computer Science and Big Data, Laureate of FST of Tangier. He is also a Business Intelligence Engineer with more than 5 years of experience in different multinational companies. The research topics of interest are smart systems, machine learning, deep learning, NLP, ANN, sentiment analysis, and smart cities. He can be contacted at email: oussama.ndama@etu.uae.ac.ma.

Mohamed Kouissi is an associate professor at the Polydisciplinary Faculty of Larache, University of Abdelmalek Essaâdi (UAE). Membre in E-DSAI2S (Research team in Data Science, Artificial Intelligence and Smart Systems), Faculty of Sciences and Technologies, Tangier, Morocco. He is an Engineer in computer Science and Specialized in software mobile development. Laureate of Tangier National School of Applied Sciences in 2011. The research topics of interest are MAS, CBR, machine learning, smart cities, and e-learning. He can be contacted at email: m.kouissi@uae.ac.ma.

Dr. El Mokhtar En-Naimi is a full professor in the University of Abdelmalek Essaâdi (UAE), Faculty of Sciences and Technologies of Tangier (FSTT), Department of Computer Sciences. (He was Temporary Professor: from 1999 to 2003 and Permanent Professor: since 2003/2004 until Now. Actually, He is a Full Professor in UAE, FST of Tangier). He was a Head of Computer Sciences Department, since October 2016 until the end of December 2020. He was responsible for a Licence of Science and Technology, LST Computer Engineering ("Licence LST-GI"), from January 2012 to October 2016. He is a Chief of Data Science, Artificial Intelligence and Smart Systems (DSAI2S) Research Team since the academic year 2022/2023. He is also a founding member of the Both Laboratories: LIST (Laboratoire d'Informatique, Systèmes et Télécommunications) Laboratory (From 2008 To 2022) and C3S (Computer Science and Smart Systems) Laboratory since the academic year 2022/2023 until Now, the University of Abdelmalek Essaâdi, FST of Tangier, Morocco. He is also an Expert Evaluator with the ANEAQ, since the academic year 2016/2017 until now, that an Expert of the Private Establishments belonging to the territory of the UAE and also an expert of the initial or fundamental formations and formations continuous at the ministry of higher education, scientific research and executive training and also at the UAE University and the FST Tangier since 2012/2013 until now. He is an author/co-authors of several articles, published in The International Journals in Computer Sciences, in particular, in MAS, CBR, artificial intelligent (AI), machine learning (ML), deep learning (DL), e-learning, MOOC/SPOC, big data, data-mining, wireless sensor network, VANet, MANet, and smart city. He was/is also Director of several Doctoral Theses in Computer Sciences. He has too served as a general chair, technical program chair, technical program committee member, organizing committee member, session chair, and reviewer for many international conferences and workshops. In addition, he is an associate member of the ISCN - Institute of Complex Systems in Normandy, the University of the Havre, France, since 2009 until Now. He can be contacted at email: en-naimi@uae.ac.ma.