Hybrid Feature Selection Framework for Machine Learning-based Bot Detection on Social Media

Amina Guendouz, Fatima Boumahdi, Mohamed Abdelkarim Remmide, Abdelghani Foura, and Amina Madani

University of Blida 1, Blida, Algeria

https://doi.org/10.26636/jtit.2026.2.2541

Abstract — Nowadays, social media impact all aspects of our lives, making us vulnerable to fraud and scams. Bots are believed to be the most prevalent form of malware that may be found in social media environments. New detection methods are required to keep up with the pace of their continuous advancement. This paper offers an overview of machine learning-based bot detection methods. The study revealed that the effectiveness of machine learning (ML) models can be significantly hindered by redundant and irrelevant features present in the datasets, which can lead to performance degradation. A hybrid feature selection (FS) combining characteristics of the genetic algorithm (GA) and the mutual information (MI) approach is proposed to overcome this challenge. The proposed method is evaluated using the following approaches: random forest (RF), decision tree (DT), support vector machine (SVM), and logistic regression (LR). Compared to the state-of-the-art models, the proposed method is capable of efficiently identifying bots using only a small number of features. For the dataset used, we achieved a classification accuracy of 0.99 using 4 features only.

Keywords — bot detection, feature selection, machine learning, social media

1. Introduction

A bot is a software tool that imitates the behavior of a real person [1]. Bots can be used for negative as well as positive purposes [2]. Social bots that perform useful services [3], such as spreading news and interacting with users, are called benign bots.

However, most bots are used to carry out malicious activities [2], [4] such as running fabricated accounts, publishing fake posts and social spam, conducting phishing campaigns, spreading rumors to manipulate people, spamming, and web scraping to steal user information. Such activities not only annoy users but also negatively impact security of the public and specific individuals.

Bot detection is relied upon to classify social network accounts as human- or bot-operated based on an analysis of their features [1]. Various techniques, such as behavior analysis-based detection systems, anomaly-based systems, graph-based detection systems, and ML-based detection systems [4] have been used for this specific purpose.

According to [5], approaches based on supervised ML algorithms are the most common and have proven to be effective under many scenarios. Nevertheless, they still suffer from some weaknesses, especially with the continuous development of bots. Existing datasets and detection approaches must keep up with this evolution to enable more effective bot-human classification.

Real-life datasets can include a wide range of features. When building an ML algorithm, we must deal with all of them, even if not all are relevant. The inclusion of unnecessary features when training a model leads to increasing the degree of complexity of the model, thus decreasing its generalization capability, and reducing its overall accuracy.

Therefore, choosing the relevant set of features used to describe the entities to be classified is a critical step in building an ML model [5]. This step, known as feature selection (FS), aims to identify the optimal set of features for building a given ML model.

In this paper, we used two different bot detection methods: the traditional one, in which classification is performed directly after data preprocessing, and the new method, in which an FS task was added preceding the classification stage. Two FS algorithms are used: genetic algorithm (GA) and mutual information (MI), in addition to a hybrid approach including both above.

For classification purposes, four ML algorithms are explored for each method: random forest (RF), decision tree (DT), support vector machine (SVM), and logistic regression (LR). Finally, a comparative analysis of the methods is carried out according to three effectiveness criteria: accuracy, precision, and F1 score.

The remainder of the paper is organized as follows. Section 2 presents existing work focusing on the detection of social media bots. Section 3 explains the outlines of the proposed approach. Section 4 offers more details concerning the contribution made. Section 5 presents and discusses the results obtained. Finally, Section 6 concludes the work and provides a brief overview of potential future paths.

2. Related Works

In recent years, significant research efforts have been dedicated to identifying bots in social media. In this study, we focus on ML models and present a review of existing work focusing on this specific field. The proposed approaches are classified according to the ML models adopted and are categorized as: supervised, unsupervised, and semi-supervised.

2.1. Supervised ML

The widest range of works described in the literature relies on supervised ML models. The authors of [6] proposed SEBD: a stream-based evolving bot detection framework that consists of three phases: data collection, streaming using Kafka, as well as detection, in which they used "Bot-MGAT" to forecast the classification of every account. In a previous work [7], they proposed the Bot-MGAT framework that combines the multiview graph attention mechanism with a transfer learning approach to identify bots using profile features only.

In [8], a graph-based X platform (formerly Twitter) bot detection HOFA framework is proposed that combats the challenge of heterophilous disguise. HOFA incorporates modules such as Homo-Aug homophily-oriented graph augmentation and FaAt frequency adaptive attention, which are based on deep learning techniques such as MLPs and attention mechanisms.

A novel bot detection model that uses personal information to construct user profiles is presented in [9]. This initiative employs advanced techniques, such as deep contextualized word embedding using ELMO Glove (global vectors) and ELMO (embedding from the language model) for the textual analysis of tweets. During the pre-processing step, the user's profile is included into all X accounts using the content of the tweets in the data. Then, an ML model is used to identify social bots by analyzing personal information.

The authors of [10] focused on the detection and classification of social bots on the X platform. They also emphasized the importance of feature engineering methods and explainable ML in improving bot detection. The authors defined new bot categories and designed two additional datasets that include accounts which have been enriched with the categories of the newly identified bots. The dataset is balanced using the ADASYN algorithm to avoid bias. Several classification algorithms have been used thereafter for binary classification and for multiclass bot detection.

In [11], a new method is proposed to encode user accounts as low-dimensional feature vectors, identifying suspicious bot accounts, and generating embeddings for information retrieval purposes. The system uses a multilingual technique to effectively detect suspicious X accounts by analyzing a set of features, regardless of the account language. The work combines relevant metadata features along with text-based features transformed into vectors independently of the language of the input text.

Paper [12] introduced a multilayer ML approach. Beginning with a dataset that has been labeled, it carries out feature extraction using standard tests and correlation analysis. To overcome the limitation correlation, the *chi2* test on non-text attributes is applied to determine 5 strongest features. Then, the text attributes undergo feature engineering, followed by an initial classification procedure that generates a vector of predictions for each text attribute. The authors provide a comprehensive assessment of the effectiveness of several classification algorithms, along with an evaluation of two widely used strategies for enhancing text attributes: bag-of-words and n-gram model.

2.2. Unsupervised ML

Only a few researchers employed unsupervised ML algorithms for bot detection. Paper [13] proposes an approach that uses unsupervised ML algorithms for bot detection on social media. Initially, a set of features is chosen to distinguish between bots and genuine accounts. Subsequently, the efficacy of two clustering techniques, namely *dbscan* and *kmean*, is evaluated on six datasets using these features. The results demonstrated that *dbscan* achieved higher efficiency by obtaining a better level of accuracy.

2.3. Semi-supervised ML

Some scientists harness the idea of using a combination of supervised and unsupervised ML algorithms for bot detection. The authors of [14] addressed the issue of classifying bots as malicious or benign. They implemented four semi-supervised ML algorithms: semi-supervised Gaussian mixture model, S3VM (semi-supervised SVM), label propagation method being a graph-based SSML model that iteratively extends the labeling of all nodes on the graph until convergence is reached, and finally label spreading (LS). They identified significant features that may be used to differentiate between benign and malicious bots and showed that SVM achieved the best results in this classification.

In [15] and [16], an approach with graph-based features, obtained from flow-level data, is presented to enhance training and inference of ML models. The proposed BotChase is an anomaly-based bot detection system that can identify bots regardless of the protocol used. It is resistant to zero-day attacks and can handle large datasets properly. The authors suggest using feature normalization (F-Norm) in addition to graph-based features in BotChase and assess other machine learning algorithms.

2.4. Semi-supervised ML

Feature selection is a task that aims to select effective subsets from original features. In machine learning, the goal of FS techniques is to find the optimal set of features allowing to create optimized ML models. The FS process eliminates irrelevant features in such a way that it reduces the dimensionality of the data, accelerates the classification process, improves the model's comprehensibility, and increases its overall performance and accuracy [17], [18]. Despite the benefits brought by FS to the ML field, their use in bot detection models is restricted. To the best of our knowledge, what we present in this section is the only existing work in this field.

In [19], four ML algorithms are tested on a public dataset, and some expressive features based on simple user profile counters are proposed for the classification of bots on X. They focus on the use of features that are easy to obtain and constitute common profile attributes, as they can be retrieved in a single request using the X API. The choice of five characteristics was determined by empirical analysis based on previous experience in developing X bots. Research emphasizes that even with a limited number of features, it is feasible to identify bots with a certain degree of complexity.

In [20], four strategies are used to identify the appropriate characteristics: correlation attributes, information gain, cross-validation attribute evaluation, and evaluation of the wrapper subset. A public dataset available from Kaggle containing 18 features is used and different ML algorithms (RF, NB, SVM, and NN) are applied to evaluate performance.

If we take an in-depth look at the existing paper, we find that, despite the good results obtained, most of these works do not attach importance to model optimization. Actual experiments are performed on large datasets that include several features.

However, not all those features are relevant for bot detection. Introducing feature selection into the bot detection process seems to be a promising initiative. By carefully choosing the most optimal features and restricting the classification task to those features only, it is possible to obtain more significant results while simultaneously reducing execution time and minimizing complexity of the system.

Nevertheless, the analyzed research fails to consider this stage or, sometimes, performs it manually.

3. Proposed Methodology

As shown in Fig. 1, we took two different paths: the traditional one (without optimization) and one relying on the proposed method (with optimization). This was done to obtain a clear picture in the comparison step, highlighting the benefits of introducing the FS task.

After importing the data, our method goes through three main stages: (i) data pre-processing, (ii) classification, and (iii) obtaining and comparing the results.

3.1. Dataset and Pre-processing

In our experiment, we used a publicly available dataset from the X platform, originating from [21]. The dataset comprises a total of 8 386 records, categorically divided into two primary groups: 3 474 records representing human interactions and 4 912 records attributed to bot activity. It contains 69 features defined for each of these accounts. These features can be categorized into three groups: content, account information, and account use features (Fig. 2).

In search of better data quality and reliability, the data from the dataset was subjected to a pre-processing step. We removed missing values, eliminated duplicates, as well as identified and handled outliers. Additionally, we standardized formats and corrected inconsistencies in the data, preparing them for accurate and effective model training.

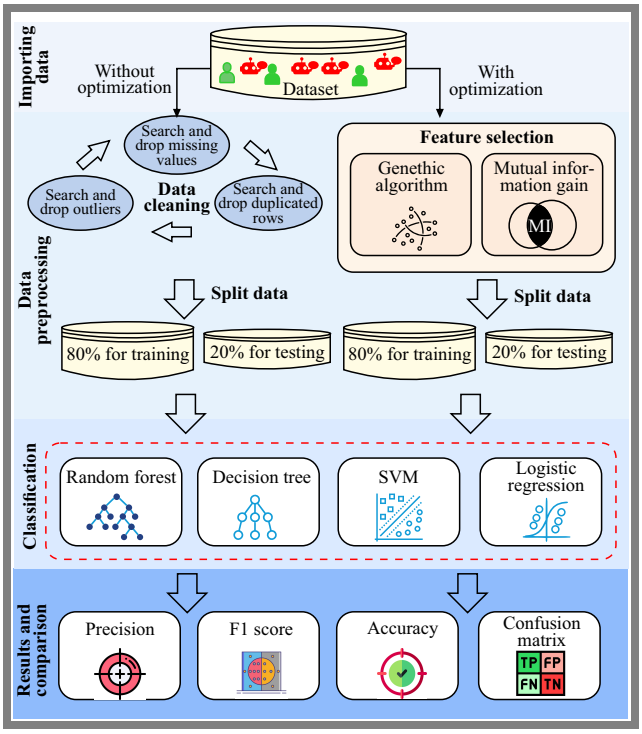

Fig. 1. Proposed bot detection methodology.

Content features	Account information	Account usage
statuses_count	ID	followers_count friends_count
favourites_count	default_profile	listed_count
min_tweet_length	default_profile_image	num_reply num_retweet min_
max_tweet_length	geo_enabled	favorite
avg_tweet_length	profile_use_background_image	max_favorite avg_favorite
min_urls max_urls	profile_background_tile	account_age
avg_urls min_hashtags	utc_offset	friends_followers_ratio
max_hashtags	protected	friends_followers_ratio_beq_50
avg_hashtags	verified	friends_followers_square_ratio
max_mentions	digits_name	2_followers_minus_friends 2_
avg_mentions	name_length screen_	followers_beq_100
max_retweets	name_length	lists_followers_ratio retweet_
avg_retweets	screen_name_length_	followers_ratio
description_length	name_length_ratio	favorites_followers_ratio
description_contains_bot	screen_name_contains_	lists_status_ratio
avg_urls_status_ratio	bot name_entropy	retweet_status_ratio
avg_mentions_status_ratio	screen_name_entropy	favorites_status_ratio
avg_favorite_status_ratio	profile_has_url	reply_status_ratio
	profile_pic_frec digits_	friends_account_age_ratio
	screen_name	followers_account_age_ratio
		favourites_account_age_ratio
		statuses_account_age_ratio
		lists_account_age_ratio
		friends_followers+friends_ratio

Fig. 2. Category of features of the X dataset (formerly Twitter).

3.2. Feature Selection and Classification

FS is introduced to find the optimal set of features that offer the best classification results. The dataset reduced to the selected features is then subjected to four classification algorithms to estimate the highest bot prediction score. We selected two algorithms (GA and MI) and used a hybrid solution comprising both.

The final stage of the proposed methodology is classification. The selected features from the previous step are the only ones considered for bot identification. The dataset is reduced to the selected feature subset. Then, it is divided into training and testing sets. We employ an 80:20 ratio to split raw and inte-

grated features, with 80% of the data set allocated for training classification algorithms and 20% for testing. The classification process involves the use of the test and train data sets for each feature subset generated from previous methods. This study uses four classifiers: random forest (RF), decision tree (DT), support vector machine (SVM), and logistic regression (LR).

4. Feature Selection Techniques

4.1. Genetic Algorithm

he genetic algorithm (GA) is inspired by the biological evolution process [22]. It is an optimization method that draws inspiration from the process of natural selection. This population-based search algorithm uses the idea of survival of the fittest. By iteratively applying genetic operators to members of the population, new populations are created. Chromosomes (population individuals), fitness function, and biologically inspired operators are key elements of GA [22]. Chromosomes are considered as possible solutions. The fitness function is used to dedicate a value to everyone in the population. After that, GA operators are applied to generate a new population.

The biologically inspired operators include selection, mutation, and crossover. Selection enables individuals to be chosen for processing in the next steps based on their fitness value. The crossover operator is a mechanism that combines two or more parents to generate new offspring solutions for the subsequent generation. During a mutation, certain pieces of the chromosomes will be randomly inverted based on probability.

The procedure of GA for FS is as follows: initial population generation, fitness function, selection, mutation, and crossover, with the next generation being produced in the final step (Fig. 3).

An initial population of solutions is randomly generated and the objective function is assessed for each member of this first generation. Chromosome genes are chosen from the dataset and without duplication. A chromosome (individual population) represents a subset of features from the original dataset. Each chromosome is a solution to the selection problem.

The fitness function serves as a tool allowing to discover the most effective features during classification (human-bot). It allows one to judge the ability of individuals (chromosomes) to survive through a fitness value and to compare them at each iteration. The fittest individuals are selected using an ML classifier. The classifiers used here are the same as those deployed in the classification phase. The fitness function computes accuracy for everyone, which represents, in this case, a feature subset. It returns the population members with the highest accuracy.

The creation of a new generation involves the selection of the fittest parents from the previous generation, followed by the application of crossover and mutation operators. The selection of individuals from the current generation, who will be the parents of the next generation, is determined randomly. However, the fittest individuals are more likely to be chosen.

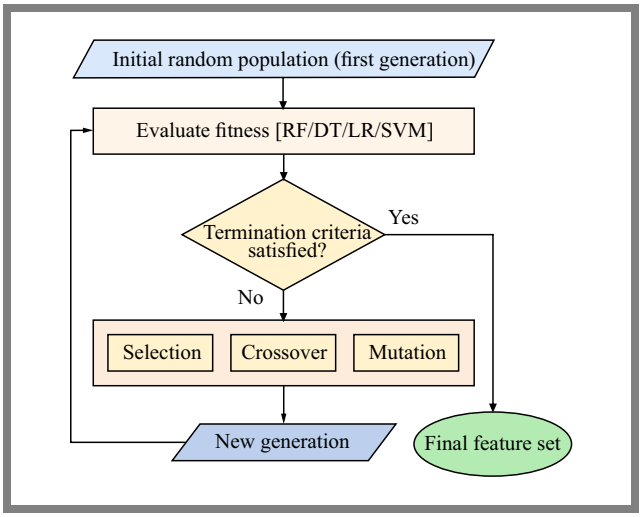

Fig. 3. Selection of features using genetic algorithm.

Suitability of a given solution is determined by its objective value, with higher objective values indicating better fitness.

A subset of the chosen solutions is utilized in a crossover operator that combines multiple parent solutions to generate new offspring solutions for the subsequent generation. The crossover operator typically produces offspring that inherit the shared traits of the parent solutions while simultaneously combining other characteristics in novel forms.

The next-generation solutions are subjected to a mutation operator, which introduces random variations within the solutions. The goal of the mutation operator is to ensure comprehensive exploration of the solution space, hence avoiding premature convergence to a local optimum.

Prior to classification, the new dataset is constructed using only the selected genes (features) from the previous step. However, once the GA converges, only the features represented by the best chromosome for a certain dataset is taken into consideration.

4.2. Mutual Information Algorithm

Mutual information relies on an elimination procedure to decrease the size of the input feature set while still preserving the discriminating class information for classification purposes. It estimates the level of information shared between two random variables. When the two variables are independent, the MI is zero. However, when the dependency of one variable on the other increases, the MI also increases [4]. In this study, the variables include both the features and the target variable (bot or not).

The formal definition of MI between two random variables is as follows:

$$MI(feature; target) = H(feature) - H(feature|target) \, ,$$

where $MI(feature; target)$ is the MI between a feature and the target, $H(feature)$ is the entropy for a feature and $H(feature|target)$ is the conditional entropy for a feature given the target.

The MI score will range from 0 to 1. A high MI value indicates a strong connection between the feature and the target,

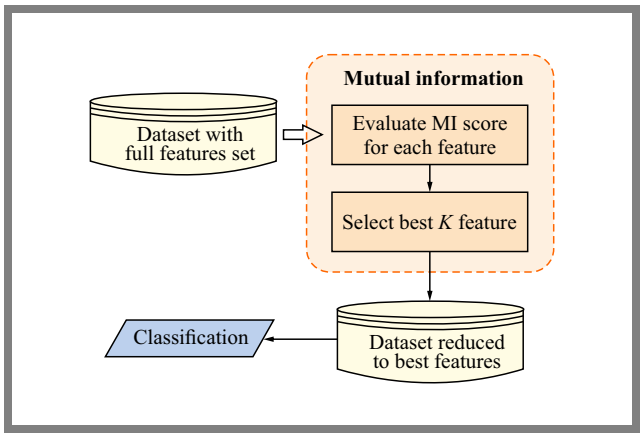

Fig. 4. Feature selection using mutual information.

highlighting the usefulness of the feature for training the model. However, a lower MI score indicates a weak correlation between the target and the feature.

Figure 4 depicts the steps taken to apply MI in the proposed method. The MI score is determined for all features in the data set. Then, a subset of k features having the highest MI score is determined. The data set trimmed to match this subset will be subject to classification in the next step.

4.3. Hybrid FS

Another experiment we have done is to perform FS using GA and MI successively (Fig. 5). GA results in a set that contains up to 30 features or more, which is still a relatively big number. We need to reduce this amount while keeping the most significant features.

Thus, after obtaining the final subset of features selected by GA, we use it as input to MI. MI then selects the best features from this subset, ensuring a more effective classification. On the one side, the number of features is reduced and on the other side, only optimal features are kept. This also has a positive influence on system complexity and classification accuracy.

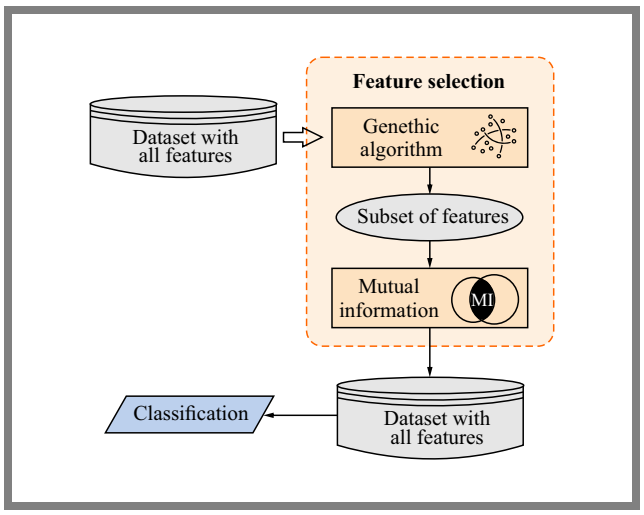

Fig. 5. Selection of hybrid features.

Tab. 1. Performance comparison.

Classifier	No. of features	Accuracy	F1 score	Precision
Without optimization				
RF		0.9505	0.9588	0.9352
DT		**0.9720**	**0.9757**	**0.9885**
LR		0.9547	0.9623	0.9390
SVM		0.9434	0.9530	0.9261
With optimization (FS using GA)				
RF	36	**0.9886**	0.9727	0.9658
DT	37	0.9833	**0.9846**	**0.9877**
LR	28	0.9696	0.9623	0.9390
SVM	38	0.9791	0.9530	0.9261
With optimization (FS using MI)				
RF	4	**0.9821**	0.9846	**0.9907**
DT	4	0.9809	0.9836	0.9866
LR	4	**0.9821**	**0.9848**	0.9788
SVM	4	0.9791	0.9821	0.9826
With optimization (hybrid FS approach)				
RF	4	0.9904	0.9918	**0.9969**
DT	4	**0.9922**	**0.9934**	0.9889
LR	4	0.9666	**0.9720**	0.9557
SVM	4	0.9755	0.9792	0.9757

5. Results and Discussion

During the experiments, we employed a new dataset described in Section 3.1 and to the best of our knowledge no previous work has used this dataset before. The use of a new dataset offers an opportunity to explore new avenues and discover bot behavior patterns that have not yet been studied. This helps foil evasion techniques developed by bot creators and improves detection efficiency.

Since this dataset has not been tested before, we chose to carry out tests without optimization before testing the proposal. Therefore, we adopted two methods for detecting bots. The first of them employed no optimization, while the other relied on an optimization method (proposed). Table 1 summarizes the results obtained by the different methods.

For the evaluation, three metrics are considered: accuracy, F1 score, and precision. Accuracy is the ratio of correctly predicted cases to the total instances in the dataset and offers a direct assessment of overall performance. F1 score is the harmonic average of accuracy and recall. It is a comprehensive statistic that considers both false positives and false negatives, thus providing a more reliable assessment of a model's performance in situations where it is important to keep a balance between identifying human and bot profiles.

Finally, precision is defined as the number of true positive occurrences divided by the sum of true positive and false positive cases. In the context of bot identification, a high accuracy value indicates that the model is efficient at reducing the occurrence of false positives. This implies that there

are fewer instances when genuine profiles are incorrectly classified as bots.

According to the literature, most of existing works do not pay attention to feature selection or rely on FS that is performed manually.

The chosen features may not be the most efficient. An automated method is needed to select the best features and test them accordingly. Ignoring this step results in more complex detection systems. Therefore, the use of algorithms for FS allows, on the one hand, to reduce system complexity and, on the other hand, to choose the most efficient features.

This work demonstrates two different algorithms for selecting the best attributes of a dataset. GA, being an optimization algorithm known for its power, and MI, which is a filter method based on computing the worthiness of each attribute. The FS step is succeeded by the classification step. During classification, four supervised ML algorithms have been tested: RF, DT, SVM and LR.

The number of features selected for each classifier for the case of GA is provided in Tab. 2. For the MI case, k is fixed at 4, so it does not change according to the classifier. The best results were obtained by MI along with the RF algorithm, reaching an accuracy value of 0.9821, an F1 score of 0.9846, and a precision result of 0.9907. Furthermore, GA with the RF algorithm achieved the values of 0.9886, 0.9727 and 0.9658 (accuracy, F1 score and precision, respectively).

In the case of the hybrid method, the features intended for MI selection are limited to the best features selected by GA. Thus, there are four cases, depending on the classifiers used in the GA's fitness function before the MI step. The selected features for each classifier along with their MI score are presented in Tab. 2. The feature IDs are also shown to establish a relationship with the charts. Figure 6 shows graphs that illustrate the classification of characteristics generated by MI of each subset, resulting from GA deployed in the previous step.

Tab. 2. Lists of selected features and MI score for each classifier.

ID	Selected features	MI score
Classifier RF		
13	max_tweet_length	0.619974
23	avg_tweet_length	0.612522
14	max_urls	0.608208
15	max_favorite	0.597547
Classifier DT		
15	id	0.617256
1	max_tweet_length	0.601209
16	min_urls	0.599594
14	max_urls	0.584531
Classifier SVM		
6	default_profile_image	0.619503
16	geo_enabled	0.611693
7	min_hashtags	0.607979
2	min_favorite	0.601555
Classifier LR		
14	max_twee_length	0.620225
15	avg_tweet_length	0.608634
4	avg_urls	0.601099
17	screen_name_length_name_length_ratio	0.598546

Results of the hybrid approach outperform all previous experiments with accuracy of 0.9904, precision of 0.9969, and F1 score of 0.9918 – with the values achieved using the RF algorithm. The results obtained with the DT ML algorithm were also significant: we achieved and accuracy value of 0.9922, an F1 score of 0.9934 and a precision result of 0.9889. In addition to the high accuracy reached, execution time and system complexity are greatly reduced, since the different classification algorithms are performed on the dataset, resulting in a restriction to four features only. On the other hand, DT and RF classifiers consistently but unevenly outperform other solutions across all four approaches.

6. Conclusions

The research revealed that the presence of irrelevant features in the datasets may degrade the efficiency of ML models, resulting in poor performance. To address this challenge, experiments have been performed in four different ways: without optimization, with optimization using GA, with optimization using MI, and by relying on a hybrid FS approach. Additionally, for each of them, four ML supervised algorithms have been tested. The results show that the hybrid method outperforms all other approaches in terms of accuracy, F1 score, and precision. The hybrid approach combines the power of the two selection methods, namely GA and MI. GA selects the best feature subset by testing accuracy of individuals by relying on various classifiers, while MI keeps only the best features according to their rank.

In future work, it will be interesting to test other FS techniques and explore a hybrid approach combining two or more techniques. It is also important to use other ML algorithms, particularly those of the deep learning variety.

References

[1] D.A. Belokurov, E.S. Shamakova, and V.S. Kolomoitcev, "Using Machine Learning Techniques to Identify Bot Accounts on a Social Network", *2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, Saint Petersburg, Russia, 2021 (https://doi.org/10.1109/WECONF51603.2021.9470605).

[2] M. Aljabri *et al.*, "Machine Learning-based Social Media Bot Detection: A Comprehensive Literature Review", *Social Network Analysis and Mining*, vol. 13, art. no. 20, 2023 (https://doi.org/10.1007/s13278-022-01020-5).

[3] Z. Ellaky, F. Benabbou, and S. Ouahabi, "Systematic Literature Review of Social Media Bots Detection Systems", *Journal of King Saud University-Computer and Information Sciences*, vol. 35, art. no. 101551, 2023 (https://doi.org/10.1016/j.jksuci.2023.04.004).

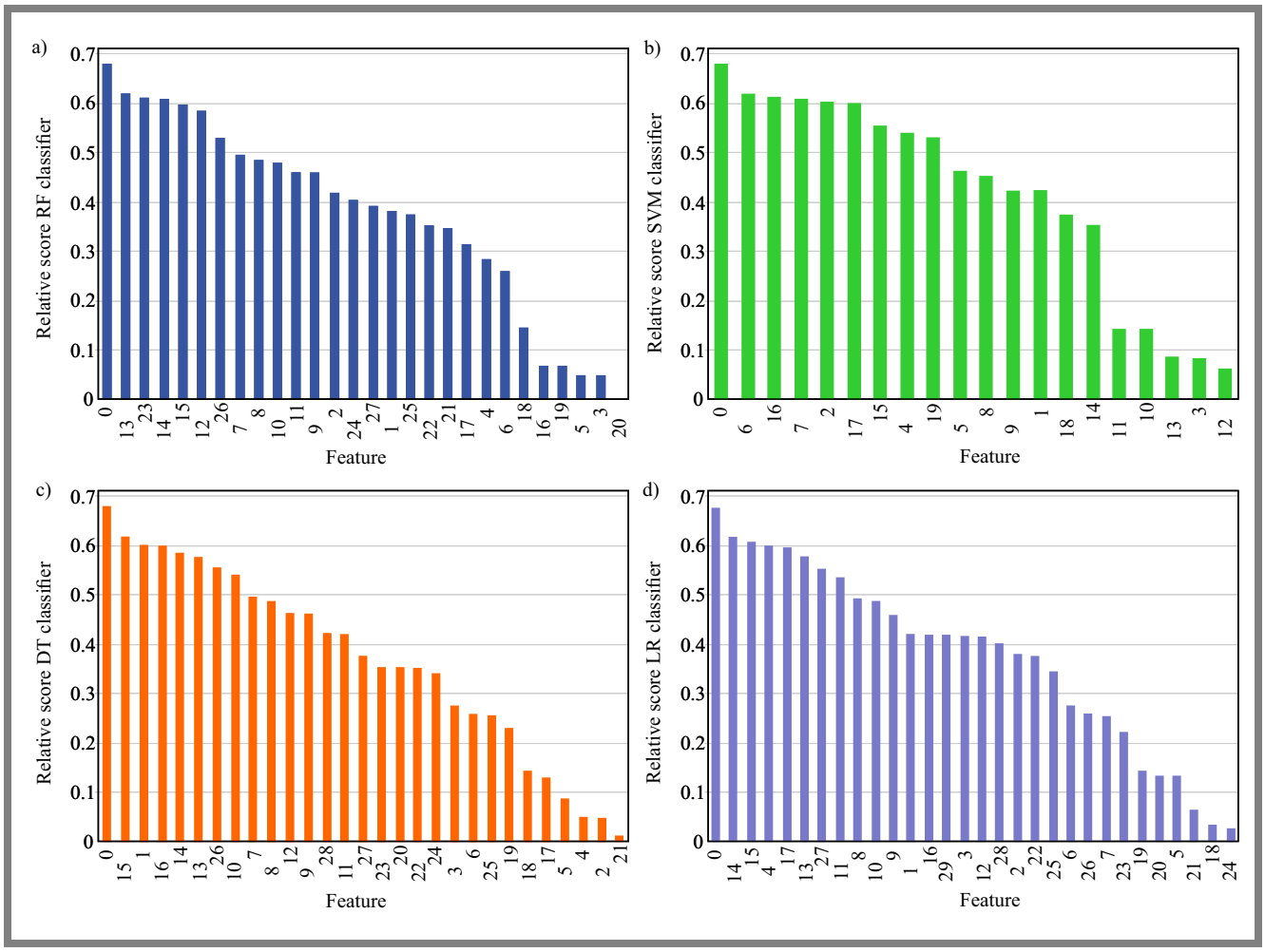

Fig. 6. MI ranking of features for each selected subset for: a) RF, b) SVM, c) DT, and d) LR classifiers.

[4] X. Wang *et al.*, "Input Feature Selection Method Based on Feature Set Equivalence and Mutual Information Gain Maximization", *IEEE Access*, vol. 7, pp. 151525–151538, 2019 (https://doi.org/10.1109/ACCESS.2019.2948095).

[5] K. Yang *et al.*, "Arming the Public with Artificial Intelligence to Counter Social Bots", *Human Behavior and Emerging Technologies*, vol. 1, pp. 48–61, 2019 (https://doi.org/10.1002/hbe2.115).

[6] E. Alothali, K. Hayawi, and H. Alashwal, "SEBD: A Stream Evolving Bot Detection Framework with Application of PAC Learning Approach to Maintain Accuracy and Confidence Levels", *Applied Sciences*, vol. 13, art. no. 4443, 2023 (https://doi.org/10.3390/app13074443).

[7] E. Alothali, M. Salih, K. Hayawi, and H. Alashwal, "Bot-MGAT: A Transfer Learning Model Based on a Multi-view Graph Attention Network to Detect Social Bots", *Applied Sciences*, vol. 12, art. no. 8117, 2022 (https://doi.org/10.3390/app12168117).

[8] S. Ye *et al.*, "HOFA: Twitter Bot Detection with Homophily-oriented Augmentation and Frequency Adaptive Attention", *arXiv*, 2023 (https://arxiv.org/abs/2306.12870).

[9] M. Heidari, J.H. Jones Jr, and O. Uzuner, "Online User Profiling to Detect Social Bots on Twitter", *arXiv*, 2022 (https://arxiv.org/abs/2203.05966).

[10] I. Dimitriadis, K. Georgiou, and A. Vakali, "Social Botomics: A Systematic Ensemble ML Approach for Explainable and Multi-class Bot Detection", *Applied Sciences*, vol. 11, art. no. 9857, 2021 (https://doi.org/10.3390/app11219857).

[11] D. Martin-Gutierrez *et al.*, "A Deep Learning Approach for Robust Detection of Bots in Twitter Using Transformers", *IEEE Access*, vol. 9, pp. 54591–54601, 2021 (https://doi.org/10.1109/ACCESS.2021.3068659).

[12] S.S. Sengar, S. Kumar, P. Raina, and M. Mahaliyan, "Bot Detection in Social Networks Based on Multilayered Deep Learning Approach", *Sensors and Transducers*, vol. 244, pp. 37–43, 2020.

[13] H. Khalil, M.U. Khan, and M. Ali, "Feature Selection for Unsupervised Bot Detection", *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2020 (https://doi.org/10.1109/iCoMET48670.2020.9074131).

[14] I. Mbona and J.H.P. Eloff, "Classifying Social Media Bots as Malicious or Benign Using Semi-supervised Machine Learning", *Journal of Cybersecurity*, vol. 9, art. no. tyac015, 2023 (https://doi.org/10.1093/cybsec/tyac015).

[15] A.A. Daya, M.A. Salahuddin, N. Limam, and R. Boutaba, "BotChase: Graph-based Bot Detection Using Machine Learning", *IEEE Transactions on Network and Service Management*, vol. 17, pp. 15–29, 2020 (https://doi.org/10.1109/TNSM.2020.2972405).

[16] A.A. Daya, M.A. Salahuddin, N. Limam, and R. Boutaba, "A Graph-based Machine Learning Approach for Bot Detection", *arXiv*, 2019 (https://doi.org/10.48550/arXiv.1902.08538).

[17] P. Dhal and C. Azad, "A Comprehensive Survey on Feature Selection in the Various Fields of Machine Learning", *Applied Intelligence*, vol. 52, pp. 4543–4581, 2022 (https://doi.org/10.1007/s10489-021-02550-y).

[18] Y. Li, T. Li, and H. Liu, "Recent Advances in Feature Selection and its Applications", *Knowledge and Information Systems*, vol. 53, pp. 551–577, 2017 (https://doi.org/10.1007/s10115-017-1059-8).

[19] J.V.F. Abreu, C.G. Ralha, and J.J.C. Gondim, "Twitter Bot Detection with Reduced Feature Set", *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Arlington, USA, 2020 (https://doi.org/10.1109/ISI49825.2020.9280525).

[20] E. Alothali, K. Hayawi, and H. Alashwal, "Hybrid Feature Selection Approach to Identify Optimal Features of Profile Metadata to Detect Social Bots in Twitter", *Social Network Analysis and Mining*, vol. 11, art. no. 84, 2021 (https://doi.org/10.1007/s13278-021-00786-4).

[21] C. Cea, "Dataset for Supervised Bot Detection on Twitter (1.0)", *Zenodo*, 2021 (https://doi.org/10.5281/zenodo.5574403).

[22] S. Katoch, S.S. Chauhan, and V. Kumar, "A Review on Genetic Algorithm: Past, Present, and Future", *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2021 (https://doi.org/10.1007/s11042-020-10139-6).

[23] M.A. Remmide, F. Boumahdi, and N. Boustia, "Toward a Hybrid Approach Combining Deep Learning and Case-based Reasoning for Phishing Email Detection", *International Journal on Artificial Intelligence Tools*, vol. 33, art. no. 2450015, 2024 (https://doi.org/10.1142/S0218213024500155).

[24] M.A. Remmide, F. Boumahdi, B. Ilhem, and N. Boustia, "A Privacy-preserving Approach for Detecting Smishing Attacks Using Federated Deep Learning", *International Journal of Information Technology*, vol. 17, pp. 547–553, 2025 (https://doi.org/10.1007/s41870-024-02144-x).

Amina Guendouz, Ph.D., Assistant Professor
LRDSI laboratory, ATM/ELT Department,
Faculty of Technology
https://orcid.org/0000-0002-7701-1336
E-mail: guendouz.amina@yahoo.fr
University of Blida 1, Blida, Algeria
https://www.univ-blida.dz

Fatima Boumahdi, Ph.D., Associate Professor
LRDSI laboratory, Department of Computer Science,
Faculty of Sciences
https://orcid.org/0000-0001-6255-9713
E-mail: f_boumahdi@esi.dz
University of Blida 1, Blida, Algeria
https://www.univ-blida.dz

Mohamed Abdelkarim Remmide, Ph.D., Assistant Professor
LRDSI laboratory, Department of Computer Science,
Faculty of Sciences
https://orcid.org/0000-0002-5145-9765
E-mail: abdelkarimremmide@gmail.com
University of Blida 1, Blida, Algeria
https://www.univ-blida.dz

Abdelghani Foura, Student
Department of Computer Science, Faculty of Sciences
E-mail: mi19.a.foura@univ-dbkm.dz
University of Blida 1, Blida, Algeria
https://www.univ-blida.dz

Amina Madani, Ph.D., Associate Professor
LRDSI laboratory, Department of Computer Science,
Faculty of Sciences
https://orcid.org/0009-0008-3896-3618
E-mail: a_madani@univ-blida.dz
University of Blida 1, Blida, Algeria
https://www.univ-blida.dz