International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 2, June 2026, pp. 477~487
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i2.pp477-487

Stacking of machine learning classifiers for bot detection using account level data

Jwala Sharma, Samarjeet Borah
Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Gangtok, India

Article Info

Article history:

Received Jan 16, 2025
Revised Nov 14, 2025
Accepted Dec 14, 2025

Keywords:

Bot detection
Feature selection
Machine learning
Social media
Stacking classifier

ABSTRACT

Social media is a platform for individuals to connect, share, and create information. Social bots produce automated content and interact with humans; in the process, they learn and mimic humans' behaviour. This research study addresses the challenge of identifying social media bots (SMB) that can rapidly disseminate information or misinformation on platforms like Twitter. It contributes to the field by reviewing literature to define bot behaviours and exploring advanced machine learning classifiers for effective bot detection using account-level data. The study employed Spearman's rank correlation coefficient to select relevant features for SMB classification, then trained six different machine learning models: decision tree (DT), random forest (RF), logistic regression (LR), support vector machine (SVM), and k-nearest neighbour (KNN). To further improve accuracy, a classifier stacking technique was applied. Key findings revealed that while individual classifiers performed variably, with RF leading at 89% accuracy, the stacked classifier approach outperformed all single-classifier methods with an impressive 90% accuracy rate. The results underscore the potential of combining multiple classifiers to enhance the precision of social media bot detection efforts.

Corresponding Author:

Samarjeet Borah
Department of Computer Applications, Sikkim Manipal Institute of Technology
Sikkim Manipal University
Gangtok, India
Email: samarjeet.b@smit.edu.in

1. INTRODUCTION

Social media is a platform where individuals, businesses, institutions, and industries share and exchange information of all kinds. It has become a medium to promote and exchange ideas, where illicit users utilize bots to promote activities of their interest. It is done by manipulating public opinions, spreading rumours, and producing fake ratings or reviews, which are auto-generated posts, comments, content, and interactions with normal users.

As per the study [1] 8.5% of users on Twitter are simply bots. Bots are responsible for disseminating a political agenda, manipulating public opinion, and boosting the voice of their propaganda in various world crisis events, such as war, natural disasters [2]. Social bots produce automated content and interact with humans; in the process, they learn and mimic human behaviour [3]. Bot accounts have become increasingly sophisticated over the years, and to some extent, they are undetected. Studies by Bessi and Ferrara [4], reveals the employment of social bots throughout the election period to enhance online network polarization by generating fake tweets, likes, and retweeting on political comments. The goal of a social bot in such an event would be either to distort the candidate's image or promote them in favour of their agenda. In 2020,

research shows the risk of social media bots (SMB) in disseminating misinformation during the time of COVID-19 pandemic [5].

As far as these challenges are concerned, a proper method must be designed to set a strict parameter that will draw a line that differentiates the real account and the bot account. Social bots exhibit various characteristics and features; understanding of each characteristic is important for the accurate classification of bots. The fundamental research question that comes across when trying to differentiate bot accounts from real human accounts is–what are the most prominent and discriminating features that help to identify the behaviour of a bot account and how it is different from a human account?

This study has investigated the features of bots and how it is different from human accounts, which is significant in classifying bots and human accounts. Contents of the paper have been organized as follows:

- The introduction section presents the background details of the research work.
- The second section presents some influential works from the literature and the identification of behavioural features of the bot.
- The third section has proposed a methodology, where a stacking ensemble classifier is implemented.
- The fourth section contains evaluation, results, and discussion. The outcome of the proposed methodology is compared with other literature review results.

2. LITERATURE REVIEW

Through various literature reviews, the behavioural features of bot accounts on Twitter have been properly identified, which could be used as a metric for differentiating between bot accounts and human accounts. To predict whether a tweet has been posted by a bot or a human account, and to enhance existing labelled datasets for social bot detection, a study has proposed “account level classification” and “tweet level” classification which resulted in area under the curve (AUC) > 98% with AdaBoost and random forest (RF) [6]. Social, demographic, and behavioural aspects of bots and humans, and the impact of bots in social environments, can be found in [7]. BotOrNot resulted in less accuracy with different thresholds (40% to 60%) in labelling an account as a bot. The presence of Twitter spambots has been evident through various analyses. An experiment was conducted using a dataset of "living, deleted, and suspended accounts" for multiple groupings of legitimate and fraudulent accounts [8]. The result of this shows that 88.9% of bots on Twitter are still alive, while only 8.6% of bots have been suspended. According to Yang *et al.* [9], tweets are retrieved from Twitter in real-time with minimal account metadata, and the result reveals that RF performed with flawless AUC when trained and tested on any single dataset.

To understand and identify spam behaviour and detect fake identities, filtering rules have been applied in [10], which resulted in successfully detecting the behaviour through sentiment analysis. The behavioural characteristics of a bot are categorized under “numeric, categorical, and series features”. Using multi naive bayes (NB), RF, and two instances of generalized linear model, an experiment was conducted which shows interesting patterns such that the popularity, follow ratio, and reciprocity bots, excluding consumption bots, have more followers than followers in general, but the case is not common in human accounts [11].

The use of emojis is popular in written communication, which plays a vital role in expressing various emotions. Emojis are used to train the model to build a sentiment classifier using MN, which determines a tweet’s emotional orientation [12]. The resultant experiments show that the model performed well in classifying positive, neutral, and negative political tweets. To identify the coordinated attempts of information dissemination, a study was conducted on a social bot, with two Bayesian statistical models, describing simple and complex contagion dynamics [13]. The result of the experiment shows that interdependent bots were more effective at spreading information than the independent bots. A [5], study highlights how SMB contributed to spreading misinformation during the COVID-19 crisis. Arista [14] used a decision tree (DT) for prediction and found that it outperformed logistic regression (LR).

With the increasing number of IoT devices, the network traffic is also an attack from bots. An efficient detection of botnet traffic by feature selection and a DT can be found in [15]. Information gain and Gini importance are used for feature selection to select the botnet traffic features that make the bots undetectable. Further, three machine learning classifiers, DT, RF, and k-nearest neighbour (KNN), are trained on the dataset and the collected set of features. The performance of the model is measured by the metric F1-score, where the DT scored the highest accuracy of 85%.

Various reviews on SMB [3], [8], [12], [16], show the presence of bot tampering with the information. Therefore, the amount of impact of social bots, which are responsible for spreading fake news and creating bias in the news over the past years, is immeasurable. As a result, it has been posing a threat to democracy and contributing to cybercrime. Not many papers have shown that the behavioural patterns of the bots are constantly evolving. In this context, the work that has been carried out in this paper evaluates the importance of features and remarkable differentiation between bots and human-operated accounts based on these features.

2.1. Behavioural features of bot account

Behavioural features play a prime role during the bot detection process. Efficient utilization of the same minimizes false positives. Most human users have their location enabled, allowing geo-location, which is one of the key signs that the account is operated by a human. The total number of tweets from bot accounts is high, and they have fewer followers compared to human accounts. Each labelled user in the database consists of profile features, which give the descriptions of their account and can be used as a metric for classification into two categories–bot and human accounts. The overall summary of features and their findings on detecting bots is shown in Table 1.

Bot accounts exhibit distinct characteristics that differentiate them from genuine users. They often lack detailed profile information and are recently created for short-term agendas, unlike legitimate accounts that show consistent, long-term activity. Bot accounts usually have few followers but follow many users to amplify their reach. They often display low follower ranking, limited likes, and excessively high retweet activity sometimes averaging up to 72 tweets per day [17] indicating automation. Their screen names are typically short or generic, and the content they post tend to be repetitive, emotionless, or filled with URLs, all of which help in identifying automated behavior on social media.

Table 1. A summary of the features used and their findings for bot detection

Approach (Ref#)	Features	Findings
[7]	Age, tweets, retweets, favorites, replies and mentions, URL count, and follower-friend ratio. (likes per tweet, retweets per tweet, user replies and mentions, activity source count, type of activity sources, and size of content uploaded	– Based on their behavioral patterns, approximately 15% of users are identified as bots. – Bots tend to be more active than human users, tweeting more frequently and at a higher rate, and bots tend to have a smaller number of followers and follow a larger number of users. – Bots use different types of content (bots being more likely to use tweet links, hashtags, and mentions) than human users. – Features, such as user activity, content, and social network characteristics, can be used to differentiate between bots and human users with a high degree of accuracy.
[8]	Features: "Fake follower frauds, retweet frauds, hashtag promotion, URL spamming, scamming, and spam of generic messages, and age of account, profile pic."	– Study provides evidence of the increasing sophistication of social spambots, which are now able to mimic human behavior and deceive even experienced users, where 8.6% are suspended bots and 88.9% are active bots.
[9]	User metadata features/Derived features "statues_count, follower_count, friend_count, favourites_count, list_count, default_count, tweet_freq, followers_growth_rate, friends_growth_rate, favourites_growth_rate, listed_growth_rate, followers_friend_ratio, screen_name_length"	– Using a large-scale dataset of Twitter accounts shows that it can achieve high accuracy in detecting social bots, even in the presence of many human users and noise in the data. – RF resulted in an AUC of 0.84.
[10]	Identity, behavior, relationship	– The RF algorithm has outperformed other algorithms used in the study while classifying between bots and humans, by achieving a high accuracy rate of over 90%.
[11]	Numeric, categorical, and series features	– There is a time variance in the tweeting period between human and bot accounts. – Miscellaneous web links and topics are included by bots than by humans.
[12]	Emoticons and emoji	– The study found that the emoji-based approach produced comparable results (F-measure = 67.8, accuracy =74.9) to more traditional sentiment analysis techniques, such as using the AFINN lexicon. – The paper demonstrates the potential of using an emoji training heuristic for sentiment analysis of social media data

2.2. Research gap

Several studies lack the necessary distinction of attributes between bot and human accounts. The present work enhances the existing literature in this field by studying behavioural patterns of bot accounts that lead to further identification of features on different levels. Previously, numerous research studies created bot detection methods utilizing single-level classifiers; however, depending on the prediction of one classifier would not be sufficient to conclude. Therefore, to improve the accuracy of prediction, a stacking approach has been employed with classifiers, integrating outputs from various classifiers and providing them as input to the final estimator for final prediction.

3. THE PROPOSED METHOD

This study proposes a classification problem to determine if a social media account is operated by a human or a bot user. The experiment has two modules–firstly, a single classifier implementation, where classifiers considered for the experiment are DT, RF, multinomial Naïve Bayes (MNB), LR, support vector machine (SVM), and KNN. A stacking classifier has been used in the second module of the experiment to improve the accuracy of the classification result.

The experiment has used several account-level features, such as: "id, followers count, friends count, listed count, favourites count, verified, status count, default profile, default profile image, screen name, location, verified". The Scikit-learn library of Python has been used for the implementation. Figure 1 shows a systematic approach for the implementation of the proposed bot detection mechanism. A publicly available dataset from Kaggle has been used to train the model. The dataset contains the classified data, which is suitable for supervised learning and training the model.

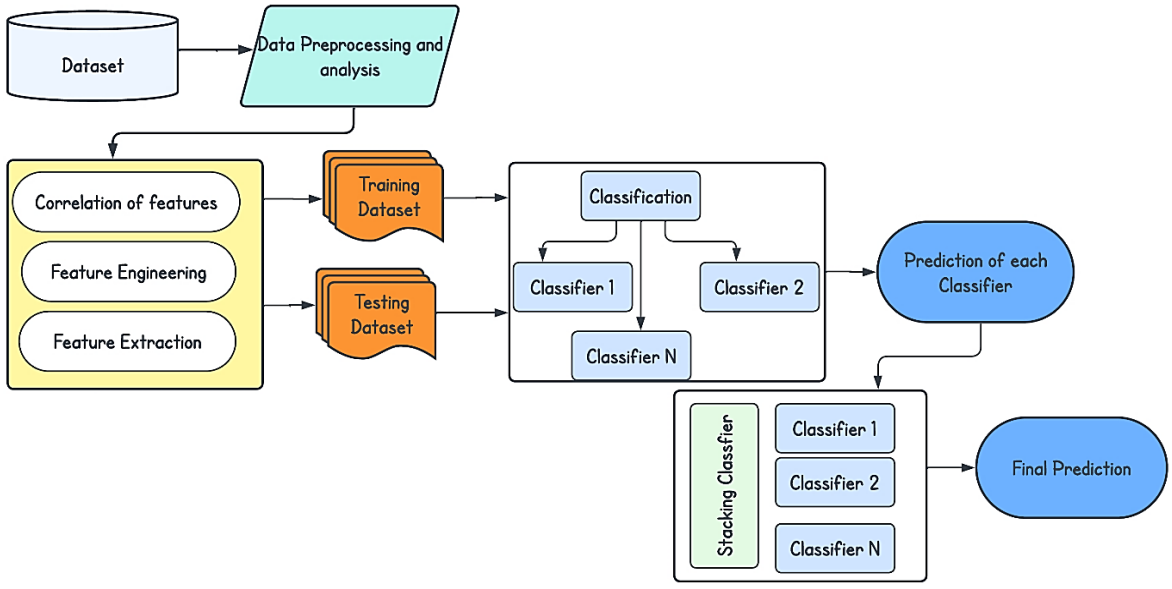

Figure 1. Methodology for bot detection

When preparing a dataset for predictive modeling, feature engineering and data preparation are essential tasks. The preparation step starts with removing elements that do not significantly contribute to the prediction goal, such as id, id_str, screen_name, and url. To validate this, the Spearman correlation coefficient has been used to check the correlation and the dependency of features from the dataset. As a result, no correlation between "id, statuses count, default profile, default profile image" and the target variable was observed, whereas a strong correlation between "verified, listed count, friends count, followers count, and the target variable" was observed.

In the second step of data preprocessing, missing values in key columns like location, description, and status are addressed to ensure data reliability. To enable compatibility with machine learning algorithms, the categorical features such as lang, has-extended-profile have been transformed into numerical representations. Additionally, numerical features are normalized to improve the performance of models like SVM and LR. In the feature engineering phase, relevant attributes, such as followers_count, friends_count, and statuses_count, are selected to focus on the most significant features. One-hot encoding has been used to ensure that the categorical features in the dataset are effectively structured for training and evaluating the model accurately.

3.1. Classifier used

Different machine learning classifiers, DT, RF, MNB, LR, AdaBoost, and SVM, have been used for similar kinds of classification problems [6]. The best accuracy result has been shown by AdaBoost and RF, with an accuracy greater than 98%. Using a multi-attribute dataset [18], a study was conducted on various machine learning algorithm that includes DT, neural networks, Kohonen maps, and correlation analysis. Using various multi-attribute dataset attributes, the proposed method is used for predicting the price segment

of real estate. The overall experiment result shows a better machine learning algorithm for predicting accuracy in terms of accuracy. The study supports that the RFC [10], SVM [6], NB, SVM, LR [12], [19], MNBC [9], RFC, and AdaBoost [9], are the best performing classifiers for classification-related problems. SVM with artificial neural network (ANN) also performs well for the detection of bots through traffic behaviour [20]. By analysing the botnet traffic and using an ensemble classifier, the study reveals that using a combinational classifier works better than a single classifier. Various influential works have depicted that SVM, DT, RF, Gradient Boosting, AdaBoost, XGB, and Extra Trees [21], [22], are the most efficient and popular machine learning classifiers for social media bot detection. The performance of the model can be enhanced with the use of various hyperparameters, such as the number of trees, depth, balancing and splitting the trees, and also with the use of data filtering and processing, an effective feature set, and employing the steps of feature engineering and feature extraction.

3.2. Algorithm

The experiment has been conducted using Python version 3.5. Scikit-learn has been used to implement different classifiers to achieve the same. The step-wise instruction has been detailed in Algorithm 1.

Algorithm 1. For stacking ensemble classifier for bot detection

Pseudocode for stacking classifier
Input:
Load and read the dataset D containing features X and target labels y
Output:
Target class (Final predicted class)

1. Convert the target variable into categorical class labels (if required).
2. Split the dataset into training and testing subsets:
   - Training data: 80%
   - Testing data: 20%
3. Define base classifiers for stacking:
   - C_1, C_2, C_3, C_4, C_5, …, C_n (e.g., SVM, KNN, Decision Tree, Naïve Bayes, Random Forest)
4. Train the individual base classifiers using the training data:
   - Train each classifier C_i using training features Xtrain and labels ytrain
   - Generate predictions on training data ($\hat{y}train_i$)
   - Generate predictions on testing data ($\hat{y}test_i$)
5. Create meta-training data for stacking:
   - Construct a meta-feature matrix using base classifier predictions:
     Xmeta_train = [$\hat{y}train_1$, $\hat{y}train_2$, …, $\hat{y}train_n$]
6. Define the meta-classifier:
   - Initialize meta-classifier C_β (e.g., Logistic Regression or Decision Tree)
7. Train the meta-classifier using meta-features:
   - Train C_β using Xmeta_train and original training labels ytrain
8. Generate meta-test data and final prediction:
   - Construct meta-test features:
     Xmeta_test = [$\hat{y}test_1$, $\hat{y}test_2$, …, $\hat{y}test_n$]
   - Predict final class labels using the meta-classifier
9. Evaluate the performance of the stacking classifier using:
   - Confusion matrix
   - Accuracy: overall correctness of predictions
   - Precision: accuracy of positive predictions
   - Recall: ability to detect all positive cases
   - F1-score: harmonic mean of precision and recall
   - ROC-AUC curve: evaluates classifier performance across thresholds
10. Save and deploy the trained stacking ensemble model.
11. End

3.3. Training of the classifiers

During the first phase of the experiment, the individual classifier is trained with the given set of data. The general structure of the training model, as in Figure 2, starts with the extraction of features (X) and labels (y) from the training data. Then, a classifier is created with specific parameters, followed by setting the parameters for each classifier.

To standardize features for better performance of multinomial, StandardScaler has been used, and the optimal value of the alpha parameter has been using GridSearchCV. Similarly, for DT and RF, GridSearchCV is used for finding the best combination of criterion, max_depth, min_samples_leaf, and ensuring the train-test split to maintain class distribution using stratify=y. The criterion ='entropy' indicates that the criteria for making a DT would be based on information gain, min_Samples_leaf indicates the minimum number of samples required, and the parameter min_samples_split indicates the minimum number required to split an internal node.

To ensure convergence for LR, the value of C has been set as 1000 and max_iter=1000. SVM kernel is configured with a radial basis function (RBF), which manages non-linear decision boundaries. The random state is set to 0, which guarantees that the output will not change even if the algorithm is executed again. Overall, for a comprehensive evaluation, a confusion matrix and, classification report have been used for better interpretability of each classifier used.

The ROC curve has been plotted for enhanced visualization for both binary and multiclass classifiers. The second phase of the experiment is followed by the stacking of the classifier, which is discussed in detail in the following section.

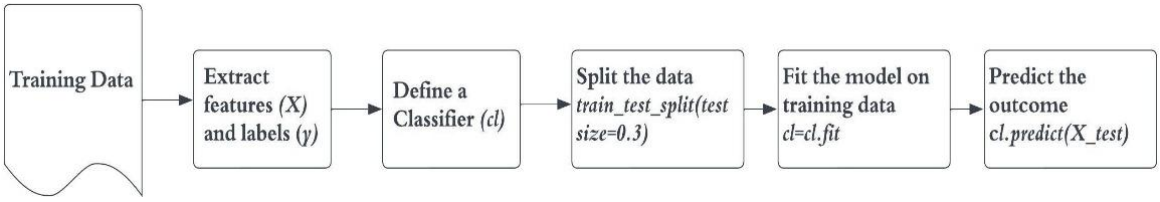

Figure 2. Methodology for training the classifiers

3.4. Creating a stacking classifier

Stacking is the simple process of combining multiple outputs from various classifiers and giving them as input to the final estimator for final prediction. Five different classifiers: DT, LR, K-neighbour classifier, RF, and SVM, are used to create a stacking classifier. In the first step, the model has been instantiated using above above-stated classifiers and defined the StackingClassifiers using LR as the final estimator. The model is then evaluated with the function evaluate_model (), which will calculate various performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC for each model. At the final step, the comparison of all individual classifiers and ensemble classifiers is evaluated, and the performance of each classifier is stored in the result dictionary(result_df); further, it can be used to create a bar plot.

4. RESULTS AND PERFORMANCE METRICS

One of the simplest forms of evaluation of the model's accuracy is by getting the accuracy score. To determine the accuracy, score total number of predictions is divided by the number of correct predictions made by the model. Mathematically, it is written as:

$$Accuracy = \frac{\text{Total no.of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

in the case of binary classification, accuracy may be assessed using positive and negative values, which can be represented as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

At different threshold levels, the Area under the curve-receiver operating characteristic (AUC-ROC) curve is used to evaluate the performance of a classification model. The ROC curve illustrates how well the model distinguishes between classes by plotting the true positive rate against the false positive rate, while the AUC value indicates the overall degree of separability between classes. A higher AUC score means the model is better at correctly classifying instances predicting 0s as 0 and 1s as 1 in binary classification tasks. Thus, the closer the AUC is to 1, the more effective the model is at distinguishing between bot and non-bot accounts.

When the ROC curve rises higher and approaches the maximum threshold value, it signifies improved classification performance. For binary classification, the AUC is calculated using the second column of the probability matrix (y_prob[:,1]), whereas for multiclass problems, the ROC-AUC score is computed using the one-vs-rest (OvR) strategy. Mathematically, which can be derived from:

$$PR = \frac{\text{TP}}{\text{TP+FN}} \quad (3)$$

True positive denotes the likelihood that the actual incident will be classed as positive. Here in this experiment, it is the probability that actual bot accounts will be classified as bots, and human accounts as human or not bots:

$$FPR = \frac{\text{FP}}{\text{TN+FP}} \quad (4)$$

in case of a false positive, it represents the probability or the measure of how frequently an actual negative instance will be classified as positive. A classification report is a performance assessment measure that gives accuracy, recall, F1 Score, and support of the classifiers used and trained during implementation.

$$Print\ (classification\_report(y\_true, y\_pred, target\_names = target\_names))$$

The target name here defines the value of predicted class, that is, in this case, either 0 or 1, and the function classification report will build a text report that shows the classification metrics. The four main metrics displayed in the classification report are precision and recall. F-1 score and support. The recall of a classifier is its ability to detect all positive events [17]. The F1 score is a weighted harmonic mean of accuracy and recall, with 1.0 being the best and 0.0 representing the worst. The number of actual instances of the class in the provided datasets is referred to as support.

The proposed stacking ensemble model outperformed all individual classifiers, demonstrating superior results across accuracy, precision, recall, F1 score, and support metrics. A 10-fold cross-validation was used to validate each classifier's accuracy. Table 2 presents the performance comparison of all classifiers, while Figure 3 visualizes their relative performance.

The stacking classifier achieved the highest overall accuracy of 0.901 when LR was used as a base classifier. Effective stacking was achieved by combining models based on multi-response linear regression and probability distributions, allowing RF, DT, LR, and KNN to complement each other. MNB was excluded due to its restriction to non-negative values.

Among individual models, RF performed best with an accuracy of 0.898, followed by KNN (0.874) and DT (0.862), while SVM and LR recorded lower scores of 0.771 and 0.758, respectively. These findings confirm the superiority of the stacking ensemble, which leverages the unique strengths of multiple classifiers to achieve enhanced robustness and accuracy.

Table 2. Summary of accuracy scores

Model	Accuracy	Precision	Recall	F1	Support
DT	0.862	0.872	0.862	0.851	840
RF	0.898	0.898	0.898	0.897	840
LR	0.758	0.758	0.758	0.758	840
KNN	0.874	0.874	0.874	0.874	840
SVM	0.771	0.771	0.771	0.771	840
Stacking classifier	0.901	0.901	0.900	0.901	840

Figures 4 and 5 display the AUC-ROC curve for individual classifiers that have been tested in the experiment and the stacking classifier. AUC measures the ability of a classifier to classify positive and negative classes accurately. The diagonal line in the figure is the performance of a random classifier, where a prediction is made randomly. From Figures 4 and 5, it can be observed that the AUC of the RF and stacking classifiers is 0.96, which indicates that they can predict the bot label and human label more accurately. Followed the performance of KNN with the AUC of 0.93, which shows good performance as compared to SVM, DT, and LR.

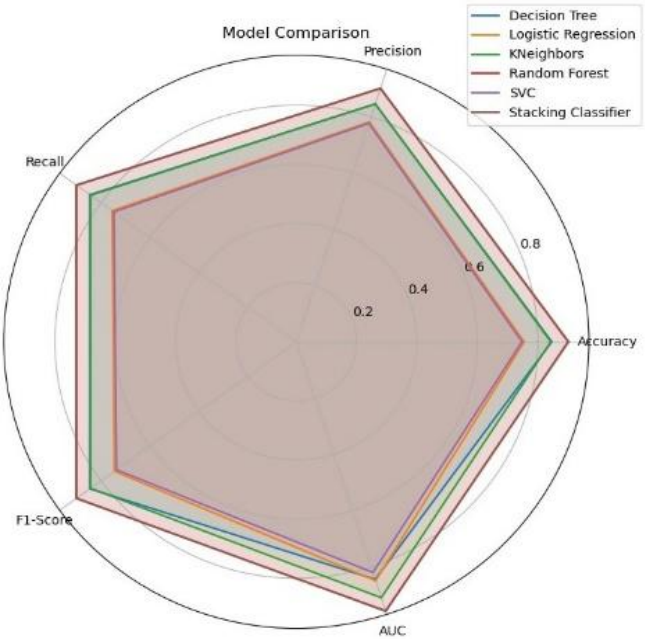

Figure 3. Performance comparison of various classifiers

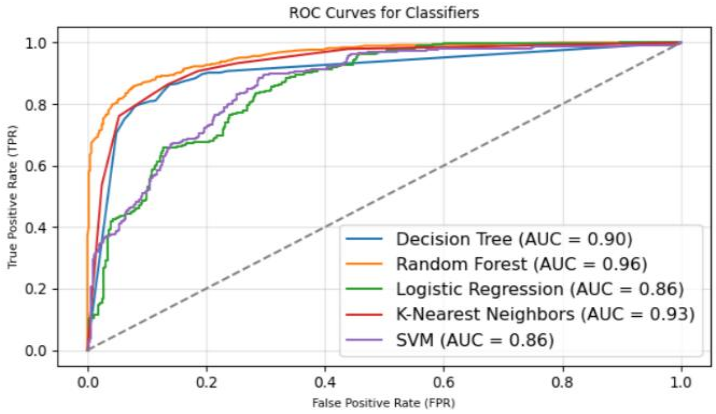

Figure 4. ROC-AUC curves of various classifiers

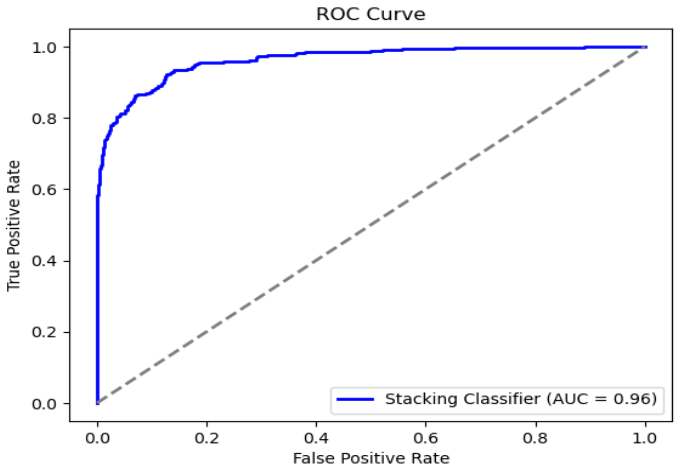

Figure 5. ROC-AUC curves of stacking classifiers

The stacking classifier effectively combines the strengths of individual classifiers, enhancing overall prediction performance. The results indicate that the RF classifier performed closely to the stacking model in terms of AUC and other evaluation metrics. However, while RF may exhibit slight overfitting during training, the stacking approach mitigates this issue by aggregating predictions from multiple models. The inclusion of a meta-learner further refines the final output, improving generalization to unseen data. Overall, stacking probabilistic models with multi-response linear regression produces superior classification results. The proposed stacking classifier demonstrates significant improvement in social media bot detection, achieving performance comparable to or exceeding that of existing state-of-the-art methods, as summarized in Table 3.

The proposed stacking classifier resulted in an outstanding accuracy of 90% across various other bot detection approaches. Very close to the proposed stacking classifier, RF has also resulted in very good performance across various metrics, which has been discussed and detailed in Table 2. The result discussed in Table 3 also emphasizes the various features used for social bot detection. The most prominent features used across various studies are account-level data and textual data, and also emoticons are used in some studies. During stacking various classifiers, the diversity of classifiers has been ensured by considering a variety of classifiers such as RF, SVM, DT, LR, and KNN classifiers. The second important factor taken care of is the choice of the meta-classifier while building the stacking classifier. The fact that LR is simple and quite effective in combining the predictions is one of the reasons to select it for a meta-classifier. To fine-tune the model, grid search has been used, and lastly, to evaluate and validate the results, cross-validation. has been performed.

Table 3. Comparison of stacking classifier with existing methods

Author/ref/year	Model/classifier	Features	Result
Lubis *et al*. [23], 2024	Deep learning-CNN	-	86.0%
Velasco-mata *et al*. [15], 2021	DT, RF, and KNN	Botnet traffic features	85% (highest accuracy of DT)
Pramitha *et al*. [24], 2021	DT, K-Nearest Neighbors, LR, and Naïve Bayes.	Text based	87.2% (highest accuracy of DT)
Mandloi and Patel. [25], 2020	Naïve Bayes, SVM, and maximum entropy	Twitter data user data and text-based	83.3%
Yang *et al*. [9], 2020	RF	User metadata features /derived features	84%
Li *et al*. [12], 2018	MNB classifier	Emoticons and emoji	74.9%
Van Der Walt and Eloff [10] ,2018	SVM, RF, AdaBoost	Identity, behaviour, and relationship	68.05%,87.11%, 85.91%
Tested	DT	Account-level data /derived features	86.2%
Tested	RF	- do -	89.8%
Tested	KNN	- do -	87.4%
Tested	LR	- do -	75.8%
Tested	SVM	- do -	77.1%
Stacking classifier [Proposed]	Stacking = {RF, DT, LR, KNN, SVM}	Account-level data /derived features	90%

5. CONCLUSION

This study helps define how bot accounts behave on social media by reviewing different research findings and comparing them with human user patterns. Bots tend to be more active than real users, posting far more frequently but having fewer followers and following many more accounts. They also use different types of content, especially links, hashtags, and mentions, making them stand out from genuine users. These patterns, along with activity and network features, can reliably distinguish bots from humans. Interestingly, studies also show that modern social spambots are becoming more sophisticated, often mimicking human behavior so well that they can even fool experienced users. Among them, about 8.6% are suspended bots, while 88.9% remain active. In the next part of this work, various machine learning models were applied to detect social bots using account-level features. The proposed stacking classifier outperformed the individual models, achieving higher accuracy, F1-score, recall, and support values. With LR as the meta-classifier, the stacking model reached a 90% accuracy rate, proving that combining multiple models and tuning their parameters can significantly improve prediction accuracy. Since social media data is complex and diverse, future work could explore more features such as URLs, hashtags, and hyperlinks from user posts. Further improvements could also be achieved by applying feature selection and hyperparameter optimization methods to make the models even more efficient and adaptable.

FUNDING INFORMATION

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the contributor roles taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P
Jwala Sharma	✓	✓	✓	✓	✓	✓		✓	✓				
Samarjeet Borah	✓							✓	✓	✓	✓		✓

C : Conceptualization
M : Methodology
So : Software
Va : Validation
Fo: Formal analysis
I : Investigation
R : Resources
D : Data Curation
O : Writing - Original Draft
E : Writing - Review & Editing
Vi : Visualization
Su : Supervision
P : Project administration
Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

On behalf of all the authors, the corresponding author states that there are no conflicts of interest.

ETHICAL APPROVAL

No Ethical approval was required for this study, since it relied on publicly available data.

DATA AVAILABILITY

The data that support the findings of this study are openly available on Kaggle. (n.d.). Twitter dataset - filtered. URL: https://www.kaggle.com/datasets/kaggledatasettbd/twitterdataset-filtered.

REFERENCES

[1] V. S. Subrahmanian *et al.*, "The DARPA Twitter bot challenge," *Computer*, vol. 49, no. 6, pp. 38-46, Jun. 2016, doi: 10.1109/MC.2016.183.
[2] S. C. Woolley, "Automating power: Social bot interference in global politics," *First Monday*, vol. 21, no. 4, Mar. 2016, doi: 10.5210/fm.v21i4.6161.
[3] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, Jun. 2016, doi: 10.1145/2818717.
[4] A. Bessi and E. Ferrara, "Social bots distort the 2016 U.S. Presidential election online discussion," *First Monday*, vol. 21, no. 11, Nov. 2016, doi: 10.5210/fm.v21i11.7090.
[5] R. Gallotti, F. Valle, N. Castaldo, P. Sacco, and M. De Domenico, "Assessing the risks of 'infodemics' in response to COVID-19 epidemics," *Nature Human Behaviour*, vol. 4, no. 12, pp. 1285–1293, Oct. 2020, doi: 10.1038/s41562-020-00994-6.
[6] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, pp. 312–322, Oct. 2018, doi: 10.1016/j.ins.2018.08.019.
[7] Z. Gilani, R. Farahbakhsh, G. Tyson, and J. Crowcroft, "A large-scale behavioural analysis of bots and humans on twitter," *ACM Transactions on the Web*, vol. 13, no. 1, pp. 1–23, Feb. 2019, doi: 10.1145/3298789.
[8] S. Cresci, A. Spognardi, M. Petrocchi, M. Tesconi, and R. Di Pietro, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in *26th International World Wide Web Conference 2017, WWW 2017 Companion*, New York, New York, USA: ACM Press, 2017, pp. 963–972, doi: 10.1145/3041021.3055135.
[9] K. C. Yang, O. Varol, P. M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1096-1103, Apr. 2020, doi: 10.1609/aaai.v34i01.5460.
[10] E. Van Der Walt and J. Eloff, "Using machine learning to detect fake identities: bots vs humans," *IEEE Access*, vol. 6, pp. 6540–6549, 2018, doi: 10.1109/ACCESS.2018.2796018.
[11] R. J. Oentaryo, A. Murdopo, P. K. Prasetyo, and E. P. Lim, "On profiling bots in social media," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10046 LNCS, 2016, pp. 92–109, doi: 10.1007/978-3-319-47880-7_6.
[12] M. Li, E. Ch'ng, A. Y. L. Chong, and S. See, "Multi-class Twitter sentiment classification with emojis," *Industrial Management and Data Systems*, vol. 118, no. 9, pp. 1804–1820, Sep. 2018, doi: 10.1108/IMDS-12-2017-0582.
[13] B. Mønsted, P. Sapieżyński, E. Ferrara, and S. Lehmann, "Evidence of complex contagion of information in social media: An experiment using Twitter bots," *PLoS ONE*, vol. 12, no. 9, p. e0184148, Sep. 2017, doi: 10.1371/journal.pone.0184148.
[14] A. Arista, "Comparison decision tree and logistic regression machine learning classification algorithms to determine Covid-19," *Sinkron*, vol. 7, no. 1, pp. 59–65, Jan. 2022, doi: 10.33395/sinkron.v7i1.11243.

[15] J. Velasco-Mata, V. Gonzalez-Castro, E. F. Fernandez, and E. Alegre, "Efficient detection of botnet traffic by features selection and decision trees," *IEEE Access*, vol. 9, pp. 120567–120579, 2021, doi: 10.1109/ACCESS.2021.3108222.
[16] F. Örnbratt, J. Isaksson, and M. Willing, "A comparative study of social bot classification techniques," 2019.
[17] O. Beatson, R. Gibson, M. C. Cunill, and M. Elliot, "Automation on Twitter: measuring the effectiveness of approaches to bot detection," *Social Science Computer Review*, vol. 41, no. 1, pp. 181–200, Feb. 2023, doi: 10.1177/08944393211034991.
[18] A. Borodulin, A. Gladkov, A. Gantimurov, V. Kukartsev, and D. Evsyukov, "Using machine learning algorithms to solve data classification problems using multiattribute dataset," *BIO Web of Conferences*, vol. 84, p. 02001, Jan. 2024, doi: 10.1051/bioconf/20248402001.
[19] L. Hagen, S. Neely, T. E. Keller, R. Scharf, and F. E. Vasquez, "Rise of the machines? examining the Influence of social bots on a political discussion network," *Social Science Computer Review*, vol. 40, no. 2, pp. 264–287, Apr. 2022, doi: 10.1177/0894439320908190.
[20] S. Saad *et al.*, "Detecting P2P botnets through network behavior analysis and machine learning," in *2011 9th Annual International Conference on Privacy, Security and Trust, PST 2011*, IEEE, Jul. 2011, pp. 174–180, doi: 10.1109/PST.2011.5971980.
[21] M. S. Kaiser, J. Xie, and V. S. Rathore, "ICT: smart systems and technologies, proceedings of ICTCS 2023, Volume 4," 2024.
[22] Z. Ellaky and F. Benabbou, "Political social media bot detection: Unveiling cutting-edge feature selection and engineering strategies in machine learning model development," *Scientific African*, vol. 25, p. e02269, Sep. 2024, doi: 10.1016/j.sciaf.2024.e02269.
[23] A. R. Lubis *et al.*, "Deep neural networks approach with transfer learning to detect fake accounts social media on Twitter," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 33, no. 1, pp. 269–277, Jan. 2024, doi: 10.11591/ijeecs.v33.i1.pp269-277.
[24] F. N. Pramitha, R. B. Hadiprakoso, N. Qomariasih, and Girinoto, "Twitter bot account detection using supervised machine learning," in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2021*, IEEE, Dec. 2021, pp. 379–383, doi: 10.1109/ISRITI54043.2021.9702789.
[25] L. Mandloi and R. Patel, "Twitter sentiments analysis using machine learninig methods," in *2020 International Conference for Emerging Technology, INCET 2020*, IEEE, Jun. 2020, pp. 1–5, doi: 10.1109/INCET49848.2020.9154183.

BIOGRAPHIES OF AUTHORS

Mrs. Jwala Sharma is a Ph.D. scholar at Sikkim Manipal University, Sikkim, India. She received her MCA degree from Lovely Professional University, and she has published one paper in Scopus and participated in three international conferences. Her areas of interest include computer network and security, data mining. She can be contacted at email: jwalachapagai@gmail.com.

Dr. Samarjeet Borah is currently working as a Professor in the Department of Computer Applications, SMIT, Sikkim Manipal University. His areas of interest include data mining, computer security, and NLP. He can be contacted at email: samarjeet.b@smit.smu.edu.in.