RESEARCH PAPER

Identification of Spambots and Fake Followers on Social Network via Interpretable AI-Based Machine Learning

[1] Dr. T. Krishnan, [2] Jujjavarapu Pradeep Kumar
[1,2]Department of CSE, KL University, Vaddeswaram, Andhra Pradesh, India
[1]Email: tkrishnan.mtech@kluniversity.in
[2]Email: pradeepkumarjujjavarapu@gmail.com

Abstract— Social networking, such as X (formerly Twitter), is the most effective way of having a large human interaction, but it is currently being overwhelmed with automated accounts that simulate human behavior and spread misinformation and manipulate opinions. The detection of such spambots is crucial in ensuring the integrity of information but many of the conventional techniques are based on opaque, black-box models that cause one to contemplate what is happening. Data used in this study is Cresci -15 and Cresci -17 to analyze interpretable ML techniques towards the identification of spambots and fraudulent followers. Both text and feature-based data are utilised and the preprocessing methods of normalisation, tokenization and removal of extraneous information are applied. RFE is a feature dimensionality reduction method that utilizes recursive selection, and resampling algorithms such as SMOTE and SMOTEENN are used to correct the issue of class imbalance. DT, RF, SVM, NB, XGBoost, AdaBoost, Stacking Classifier, and Voting Classifier are some of the various ML techniques evaluated. The findings show that the Stacking Classifier is highly accurate, achieving 99.9% on the Cresci-15 dataset and 99.5% on the Cresci-17 dataset. Moreover, explainable AI models such as LIME and SHAP allow one to visualize the importance of each feature, thus enhancing model transparency and making it easier to make decisions. These results emphasize the effectiveness of incorporating the feature selection method, advanced resampling as well as ensemble learning techniques along with the interpretable means to the dependable determination of the automated accounts in the social networks.

Keywords— Interpretable AI, social network, bot detection, fake followers, spambots.

I. Introduction

The social networks have turned out to be the major source of information transmission in the modern digital era. X (formerly Twitter) has become one of the most recognizable and influential platforms that have allowed creating real-time connections and interactions between active millions of users [1]. The social and economic consequences of it are considerably far-reaching, but also attracted dangerous personalities who use its loose nature to influence the masses and spread fake news. One of the most common tools used towards this goal is automated applications also known as bots. Even though some bots are beneficial in that they generate useful information such as teaching blogs and news posts, evil bots are created to propagate spam, bad information, and harmful content [2].

The discovery of these illicit personalities has become a major challenge in the preservation of integrity in the online discussion. User metadata, behavioral features, timestamps, and network structures are the most common methodologies of determining bot detection [3]. Nevertheless, these attributes are not just made automatically and require a significant amount of effort and expertise to be designed and extracted. Moreover, bots continue to develop, and they adopt adaptive strategies that mimic the behavior of humans, which makes them increasingly more difficult to detect. Bad bots make it that much easier to proliferate fake news, hate speech, and other misinformation through systematic interactions with high-end accounts [4].

Botnets and Sybil accounts also contribute to the worsening of the environment of automated manipulation. The group of synchronized bots performing specified malevolent actions is referred to as a botnet, but Sybil identities are pseudonymous identities utilized to control legitimate conversation. These methods are detrimental to genuine interaction and worsen the spread of low-credibility information, therefore, creating a major issue of trust and reliability in social media ecosystems. Against this background of such risks, ML has been widely applied to sports analytics [7], sentiment analysis [8], [9] and locating bogus news [10]. Social bot detection has been addressed by ML-based methodologies in the last few years, which have offered automated solutions in comparison to traditional heuristic and network-based methods. However, a few ML-based detection models still remain as black-box models, that is, with limited interpretability. Such opens the possibility that researchers and practitioners struggle to determine the extent to which predictions reflect the existence of meaningful patterns or the overfitting of noises.

Interpretable ML (XAI) is increasingly being integrated into bot detection systems in order to reduce these limitations. XAI increases transparency through explaining the effect of single attributes on model predictions, and thus, increasing accountability and confidence. XAI-based systems improve reliability and interpretability of algorithmic decisions, making it easier to create more reliable and trustful systems used to detect bots on X and similar social networks.

II. Related Work

The detection of bots through social networks has become an essential area of study due to the dynamism in the nature of fraudulent accounts that pervert internet conversation and information frameworks. Scholars have successively explored

advanced ML, graph-based methods, and explainable models to support accuracy of detection and scalability.

Mbona and Eloff [11] pointed out that it is important to classify bots as potentially dangerous and innocent, since many bots play neutral or even positive purposes, including the distribution of news. They introduced a semi-supervised learning approach where very little labeled data are used and a lot of unlabeled data is used data that reduces the need of costly annotations, and enhances a semi-supervised learning method that implements very little labeled data and a lot of unlabeled data. Similarly, Yang et al. [12] addressed the scalability and generalizability problem by suggesting a data selection approach that ensures the functionality of ML models trained on representative data sets across a wide range of settings, which is necessary to create versatile detection systems.

Generative bots that are AI-driven pose new challenges. Lopez-Joya et al. [13] analyzed the structure of bots trained on big language models and show that they are harder to detect since they are able to generate realistic and contextually coherent text. They emphasized the fact that the new approaches have to be reviewed to be able to adjust to the AI-enhanced competitors. Also, an elaborate model combining a pre-trained auto-encoder with GNNs was presented by Pham [14], which would prove highly beneficial in terms of capturing latent behavioral traits as well as structural social relationships. This composite approach promoted cross-network generalizability, which was not available with conventional methods of supervision.

As Terumalasetti and Reeja [15] emphasised, it can be necessary to enhance the user trust by developing a multi-dimensional analytics system that combines content, temporal, and relational features. They had a sophisticated architecture that was resistant to adaptive spambots, and they needed multidimensional models rather than rely on single feature types. Paudel et al. [16] had previously demonstrated how the use of bot tactics had evolved to imitate human-like interactions in networks. Using advanced network analysis, they showed that such structural measures as clustering coefficients and centrality measures are useful indicators of advanced bots.

Aljabri et al. [17] also reviewed the progress in the field through a comprehensive literature review of ML-based bot detection. Among the issues that were highlighted in them were dependence on feature engineering, inflexibility in scalability, and low transparency of black-box models. In their review, they noted that it was important to have a balance between detection accuracy and interpretability as well as adaptability. Wu et al. [18], further extended this perspective with Botshape which is a paradigm aimed at studying behavioral patterns, such as the frequency and the types of posts temporal activities. Their method involving the simulation of dynamic behaviors showed strong performance compared to bots that are designed to imitate human timing.

Graph based methodology has also received attention. To reduce reliance on manually produced characteristics, Bebensee et al. [19] proposed to use node neighborhoods and egograph topology to determine relational differences on the micro-structural level of social networks. Their study proved that graph-based learning is able to detect anomalies within contextualised network structures. Dimitriadis et al. [20] revealed Caleb, a conditional adversarial learning system that increases resiliency to adaptive bot methods. Instead, their plan enhanced generalization and permanence by training on adversarial cases, having a first hand experience on the arms race between detection systems and attackers.

III. Materials And Methods

The proposed approach describes a powerful and explainable ML model to detect social spambots and fake followers on social networks by utilizing feature-based synthesis and text-based data of Cresci-15 dataset and Cresci-17 dataset. The process of data preprocessing includes handling the parameters of missing values, the encoding of discrete variables, the standardization of numerical features, text cleaning (removal of HTML tags, URLs and non-printable characters), normalization, tokenization, and TF-IDF representation. Class imbalance is fixed using SMOTE and SMOTEENN. A predictive model is built on the basis of DT, RF, SVM, XGBoost, NB, and AdaBoost and Stacking and Voting Classifiers are introduced. Recursive Feature Elimination is used for feature dimensionality reduction, while LIME and SHAP are employed for model interpretability. An interface based on a Flask helps to detect and study model decisions in real-time and interactively.

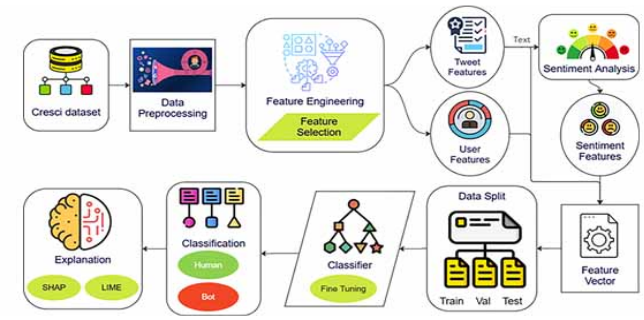

Fig. 1. System Architecture

The proposed system organization begins with the Cresci data which is followed by data preprocessing to clean the data. They use feature engineering and selection to come up with meaningful tweet, user and emotion features to be represented. The processed data is divided into three categories i.e., training, validation, and testing. These sets develop feature vectors which could be utilized in the modeling process. Different fine-tuned classifiers are utilized to distinguish human accounts and bot accounts. Explainable AI, like SHAP and LIME, can be described as being transparent, since they explain the decisions made by the model, thus adding to trust and interpretability in bot identification.

a) Dataset Collection:

i) Cresci-17: The Cresci-17 big data was compiled to analyze social spambots, traditional spambots, cheated followers, and genuine accounts on Twitter. There are 14,368 users and 6,637,616 tweets, 10,894 bots and 3,474 humans at the user level, 3,798,254 bots generated tweets and 2,839,362 human generated tweets. Different classes of spambots, e.g., social spambots, traditional spambots, and fake followers, were included. Data on user profiles and tweets were mined, combined and labeled, and produced a labeled dataset suitable for training ML and explainable AI-based bot recognition systems; however, the class distribution was imbalanced and required resampling during preprocessing.

	user_id	followers_count	friends_count	statuses_count	favourites_count	listed_count	verified	default_profile	created_at	account_age_days	description	description_len
0	1502026416	208	332	2177	265	1	0	0	2013-06-11 11:20:35	4474	15years ago X.Lines24	
1	2492782375	330	485	2660	3972	5	0	1	2014-05-13 10:37:57	4138	[illegible]	
2	293212315	166	177	1254	1185	0	0	0	2011-05-04 23:30:37	5243	Let me see what your best move is!	
3	191839658	2248	981	202968	60304	101	0	0	2010-09-17 14:02:10	5472	20. menna: #farida #nyc and the 80s actually y...	
4	3020965143	21	79	82	5	0	0	1	2015-02-06 04:10:49	3870	Cosmetologist	

Fig.2 Cresci-17 dataset

ii) Cresci-15: Cresci-15 dataset combines five twitter data sets, namely: E13, TFP, FSF, INT, and TWT, of 5,301 individuals and 2827,758 tweets. It has 1,950 human accounts and 3,351 bot accounts. All user profiles consist of 35 raw attributes such as follows, friends, statuses, favorites, number of lists, status of verification, profile information, and date of account creation. Textual features such as as word count, hashtags, mentions, URLs, emoticons, punctuations, sentiment, and engagement analytics are added to tweets. On average user-level features create 29 complete variables hence making it easy to detect bots and analyze their behaviors.

	user_id	followers_count	friends_count	statuses_count	favourites_count	listed_count	verified	default_profile	created_at	account_age_days	description	description_length
0	3610511	5470	2385	20370	145	52	0	0	2007-04-06 10:58:22	6733	Founder of http://www.screenweek.it & http://w...	151
1	5656162	506	381	3131	9	40	0	0	2007-04-30 15:08:42	6709	BSc degree (cum laude) in Computer Engineering...	104
2	5682702	264	87	4024	323	16	0	0	2007-05-01 11:53:40	6708	Cogito ergo bestemmio.	22
3	6067292	640	622	40586	1118	32	0	0	2007-05-15 16:55:16	6694	Se la vita ti dÃ sarde, scapocciale!	37
4	6015122	62	64	2016	13	0	0	0	2007-05-13 19:52:00	6695	Je me souviens	14

Fig.3 Cresci-15 dataset

b) Pre-Processing:

The following units of the preparation procedure were applied to the data sets Cresci-15 and Cresci-17, and prepared, converted, and optimized user and tweet information were applied systematically to ensure valid and accurate classification of bot and human accounts.

1. Data Processing: The raw user and tweet data of the Cresci-15 and Cresci-17 datasets were amalgamated between all the sub-datasets. Missing numerical values such as follows, friends, statuses and favorites were replaced with zeros. The dates of creating the accounts were converted to the datetime format and account age in days had been calculated. Categories were normalized (0=human, 1=bot), which ensured that the same occurred in both datasets in terms of accounts being identified in the same way and then feature engineering and model training occurred.

2. Feature Engineering: The data of users and tweets were transformed to derive noteworthy features. The users were calculated on metrics like the number of followers/friends, the length of description, whether they have a description, and the sentiment scores. The attributes at the tweet level included words counts, hashtags, mentions, links, emojis, punctuations, distinctive word percentage, capitalization percentage, duration of the tweet, and sentiment analysis. The numbers of hashtags, mentions, emojis, engagement, and sentiment per user were aggregated and produced an extensive set of features to be used in modeling.

3. Visualization: The distribution of features of the two sets of data was studied to gain further knowledge on trends, ranges and variation. The numerical variables, such as follows, friends, length of tweets, and age of the account, were presented in the form of histograms, boxplots, and scatter plots. The outliers, skewness distributions and outliers between people and bots were found in the visual analysis. This assisted in processing and standardizing data and feature selection and model construction.

4. Feature Selection: The quantitative and bulk characteristics of tweets were optimized to ensure the process of prediction was as accurate as possible. Duplicates and those sections that contained minimal information were removed. It monitored such information as the number of profile-based variables: the number of friends, the tone of the descriptions, the mean number of hashtags, the number of mentions as well as the number of followers the emoticons, the activity, and the analytics of tweets. The selection of the features also ensured that models paid attention to the most valuable attributes. This contributed to their increased effectiveness, reduced noise and enhanced the categorization of both sets of data.

In order to validate the effectiveness of the Recursive Feature Elimination (RFE) technique, an experimental comparison was conducted before and after applying RFE. Initially, a total of 48 engineered user-level and aggregated tweet-level features were used for training the classifiers. Using Logistic Regression as the base estimator, RFE selected the top 20 most informative features by recursively eliminating weakly contributing variables.

Table.1 Performance comparison before and after RFE

Feature set	No. of features	Cresci-15 Accuracy	Cresci-17 Accuracy
Before RFE	48	99.9%	99.5%
After RFE	20	99.8%	99.4%

To evaluate the impact of RFE, the Stacking Classifier was trained using the complete feature set and the reduced feature set obtained after RFE. The experimental results show that feature reduction preserved classification performance while reducing feature dimensionality and model complexity. This confirms that RFE effectively removes redundant and less informative attributes while maintaining high detection accuracy for spambot and fake-follower identification.

5. Data Balancing: There was a power mismatch between the data sets with the bots being more numerous than human accounts. In order to rectify this and ensure that both real and fake accounts were duly reflected, over sampling of the minority class or fabrication of fake data was done. In fair datasets, predictive models are not as biased on the majority class, and they can make predictions. The features at the user level were assembled in a manner that they were consistent, and fair training data was employed to ensure that the classifications were accurate both in Cresci-15 and Cresci-17.
6. Scaling: Numerical variables such as followers, friends, statuses, favorites, tweets and interaction scales were normalized using the Min-Max or z-score. Making the range of features normal reduced the effect of the outliers and ensured that all the features were well represented when training. With these developments, gradient-based classifiers became more precise; hence, predictions of these anticipated features were more accurate. It facilitated the comparison of Cresci-15 and Cresci-17 data in features.

C) Training and Testing:

Stratified sampling was used to create a training and testing set dividing up into the processed and feature-engineered data to maintain the class distribution. Aggregated tweet features and user level features were used to train supervised classifiers. Accuracy, precision, recall and F1-score were among other things that the models were evaluated. This allowed the Cresci-15 and Cresci-17 sets to generalize quite effectively, and produced correct classification of bot and human accounts using numerous diverse features. All experiments were conducted at the user-account level after aggregating tweet-level features into comprehensive user representations for classification.

D) Algorithms

Decision Tree: DT method considers both the feature based and text qualities, in a technique that recursively divides the data based on the most information rich features. It provides a straightforward structure and enables an easy categorization of accounts and also competently classifies category and numerical information to identify spambots and fraudulent subscribers.

$$I(i) = 1 - \sum_{i=1}^{k} p_i^2 \quad (1)$$

Random Forest: RF runs many DT and summarizes their output through majority voting, therefore, improving noise and overfitting resistance. It simultaneously evaluates multiple variables, which increases generalization and accuracy and closely identifies automated accounts with nonlinear behavior in social network settings.

SVM: SVM finds a suitable hyperplane to maximize the distinction of classes between human and automated accounts. It owes its effectiveness to high-dimensional analysis with the aid of kernel functions that make it easy to capture subtle differences in features without compromising its strength against overfitting and noisy data.

XGBoost: XGBoost is a sequentially boosted DT that is trained using gradient optimization, which is a more effective way of adjusting to complex feature interactions and more accurate than traditional models. It is fast at processing structured and unstructured attributes, can scale to large datasets, imbalance can be effectively handled and it provides sound performance in spambot detection.

$$\hat{y}_i = \sigma\left(\sum_{k=1}^{K} f_k(x_i)\right), f_k \in F \quad (2)$$

Naïve Bayes: Naive Bayes is a type of categorization, which relies on probability, and uses conditional independence of features. It is effective in analyzing large volumes of text and structured data and classifying accounts based on probability. Even though it is simple, it offers efficiency which serves as a lightweight yet reliable base of bot identification operations.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad (3)$$

AdaBoost: The AdaBoost algorithm identifies such types of false classifications and assigns them various weights to enhance the performance of detecting them using sequential weak classifiers. This method makes it possible to have more accurate predictions, reduced bias, and correct imbalances. This implies that a hybrid model can be developed which can detect spambots in large datasets.

Stacking Classifier: In stacking, you combine a number of base models that are trained with the output of the other models. It takes much of the advantage of algorithms, reduces a portion of the weaknesses, and increases the overall detection rates, scanning both features and text characteristics. This is to ensure that automated account detection and fraud cases are not left behind.

$$\hat{y} = g(Y_{base}) = g\left(f_1(x), f_2(x), \ldots, f_m(x)\right) \quad (4)$$

Voting Classifier: Voting involves the outcome of multiple classifiers and is a decision based on a majority vote. This renders the decisions more solid and reduces the errors of individual models. It employs two dissimilar types of algorithms operating simultaneously to identify spambots and fake followers in a just and appropriate manner. These findings are also understandable.

$$\hat{y} = \text{argmax}_c\left(\sum_{i=1}^{n} II\left(\hat{y}_i = c\right)\right) \quad (5)$$

E) Integration of XAI and Flask Framework:

Explainable Artificial Intelligence (XAI) and Flask framework will ensure that ML applications are open-source and accessible. Such XAI methods as SHAP and LIME can assist us in the interpretation of how the results of a model are explained by considering the functions of features and decision-making mechanisms. It is simpler to add such explanations so that the customer can consider the reason why

the model would refer to an account as a bot or a human which increases credibility and responsibility.

Flask is an extremely aggressive python web-based system, which allows them to utilize these models and visual explanations. It also simplifies the process of the AI models in the backend to communicate with the user experience in the frontend. One can make decisions, obtain predictions, and observe the functioning of XAI generated answers in real-time, which is why this system is not complicated to operate, understandable, and user-oriented.

IV. Experimental Results

Accuracy: Accuracy is the proportion of correctly classified instances (bot and human accounts) among all predictions made by the model. It is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (6)$$

Precision: Precision measures the percentage of correct cases of the positive cases identified. As a result, the precision formula can be described to be:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (7)$$

Recall: The machine learning measure, recall, informs you of how effective a model is at classifying all the pertinent examples of a particular class. It is the ratio between the number of correctly identified positives and the actual positives. It reports the capability of a model to identify the instances of a particular class.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (8)$$

F1-Score: F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure when class distribution is uneven. It is calculated as:

$$\text{F1 Score} = 2 * \frac{\text{Recall X Precision}}{\text{Recall} + \text{Precision}} * 100 \quad (9)$$

Table.2 Performance Evaluation – Cresci – 15

ML Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.979	0.979	0.979	0.979
RF	0.993	0.993	0.993	0.993
SVM	0.988	0.988	0.988	0.988
XGBoost	0.995	0.995	0.995	0.995
Naïve Bayes	0.938	0.943	0.938	0.938
AdaBoost	0.994	0.994	0.994	0.994
Stacking Classifier	0.999	0.999	0.999	0.999
Voting Classifier	0.998	0.998	0.998	0.998

The comparative results given in Table 2 show that overall performance of Stacking Classifier was the best of all the methods evaluated.

It is observed that accuracy, precision, recall and F1-score obtain identical values for several models because the evaluation is performed on a balanced test split and the predicted class distributions closely match the ground-truth labels, leading to identical aggregated performance measures.

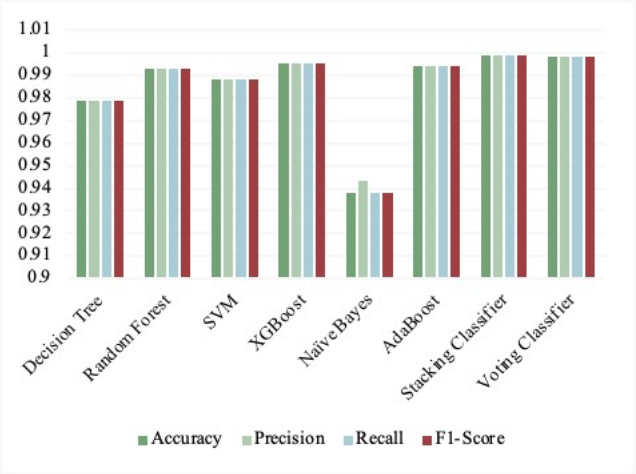

Fig.4 Comparison Graph – Cresci – 15

As it is presented in Figure 4, the Stacking Classifier is the most successful model, and the Accuracy is represented through green Figure, Precision through dark green, Recall through light blue, and F1-Score through red on all models.

Table.3 Performance Evaluation – Cresci – 17

ML Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.975	0.975	0.975	0.975
RF	0.990	0.990	0.990	0.990
SVM	0.954	0.955	0.954	0.954
XGBoost	0.991	0.991	0.991	0.991
Naïve Bayes	0.689	0.799	0.689	0.657
AdaBoost	0.987	0.987	0.987	0.987
Stacking Classifier	0.995	0.995	0.995	0.995
Voting Classifier	0.987	0.988	0.987	0.988

According to Table 3, the Stacking Classifier achieved high overall performance in comparison with other models of ML evaluated.

It is observed that accuracy, precision, recall and F1-score obtain identical values for several models because the evaluation is performed on a balanced test split and the predicted class distributions closely match the ground-truth labels, leading to identical aggregated performance measures.

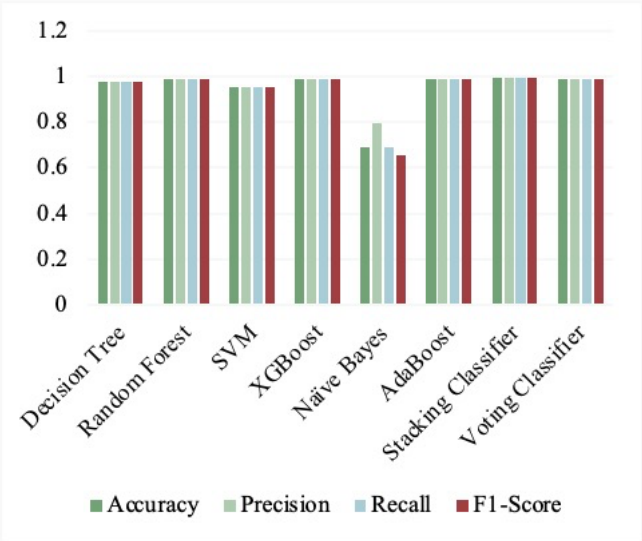

Fig.5 Comparison Graph – Cresci – 17

Figure 5 is a representation of the Stacking Classifier algorithm which is the most effective algorithm, comparing the eight classifiers on the four scores: Accuracy (green), Precision (light green), Recall (light blue), and F1-Score (brown).

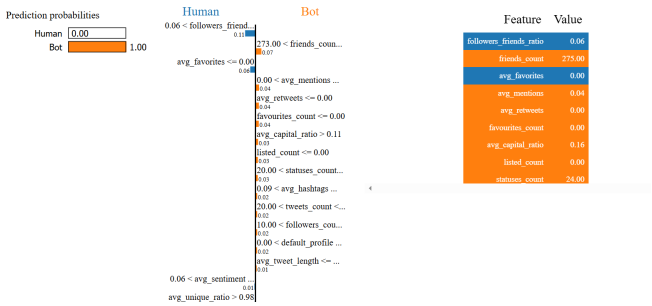

Fig.6 LIME

Fig.6 LIME visualizes local feature contributions explaining individual bot prediction decision transparently.

Fig.7 SHAP

Fig.7 SHAP summary plot shows global feature importance influencing overall bot classification performance.

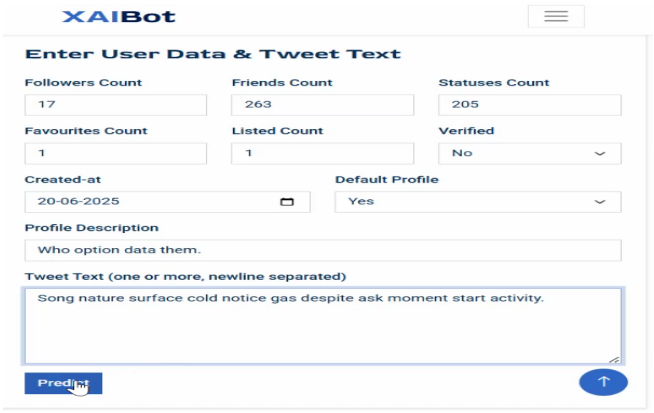

Fig.8 User profile & tweet input interface

The interface shown in figure 8 is used by the users to input their profile details and tweet content which include the number of followers, friends and the updates on their status.

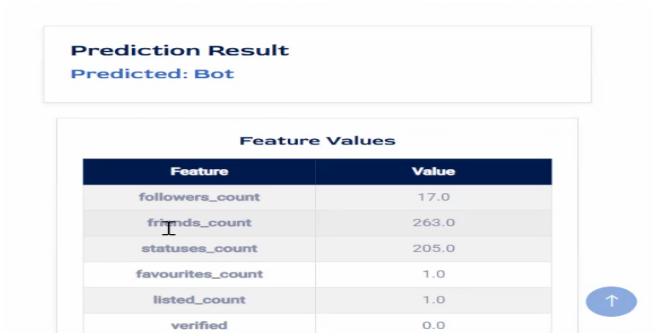

Fig.9 Predicted Results

The output of the prediction is given in Figure 9, which refers to Bot and the corresponding values of the feature provided to be analyzed and classified.

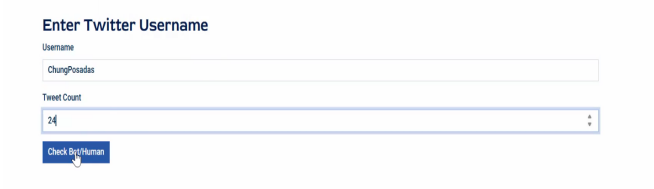

Fig.10 Username-based account checking interface

The interface presented in Figure 10 gives the user the option to type in a username to determine whether the account is run by a bot or a human.

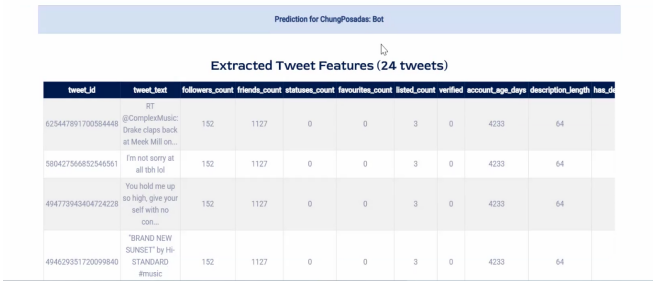

Fig.11 Predicted Results

The result of prediction is shown in figure 11 and it is Bot and the relevant account feature information.

V. Conclusion

The study shows that a combination of feature-based and text-based data and complex ML algorithms can effectively detect spambots and fake followers on the social networks. By employing Recursive Feature Elimination (RFE) in reducing features and alleviating the problem of class imbalance with SMOTE and SMOTEENN, the models were able to achieve better predictive performance. The best results were obtained using ensemble learning methods, which included the Stacking Classifier, with 99.9 percent accuracy on the Cresci-15 dataset and 99.5 percent accuracy on the Cresci-17 dataset, thus supporting the effectiveness of using a large number of methods. The integration of explainable AI techniques, including LIME and SHAP, provided in-depth information on the importance of features, which can guarantee the introduction of model interpretability, and high accuracy is achieved. The results confirm that complex ensemble-based models, which make use of carefully selected features and balanced data sets, can significantly improve the automation account detection, but interpretable models improve the understanding of the decision-making mechanisms. The results show that a combination of feature selection, sampling techniques, and explainable ML algorithms can improve the accuracy of detection and transparency and reliability, so as to provide a comprehensive solution of tracking and reducing the impact of spambots and fake followers in social networks. Further studies can be focused on the expansion of the detection framework to support the dynamics and more sophisticated automated accounts that can constantly modify their behavior to evade the detection. It should be scalable and react to spambot activity in real time, which would be enhanced by integrating real-time data feeds of most social networks. Stated in simple terms, advanced DL techniques, including transformer-based language models can be explored in order to identify complex text-based patterns alongside feature-driven indicators. Moreover, multimodal data, such as photos, videos, and network interactions, would be useful in improving the precision of detection. Further research on explainable AI schemes will lead to greater transparency of more and more complex models, where users and administrators can be able to clearly understand predictions. Improving computational efficiency to be deployed in large-scale social network environments will become a key area of improvement.

References

[1] Alkathiri, N., & Slhoub, K. (2025). Challenges in machine learning-based social bot detection: a systematic review. Discover Artificial Intelligence, 5(1), 214.

[2] Akhtar, M. M., Bhuiyan, N. S., Masood, R., Ikram, M., & Kanhere, S. S. (2025). BotSSCL: Social Bot Detection with Self-Supervised Contrastive Learning. Online Social Networks and Media, 48, 100318.

[3] Lopez-Joya, S., Diaz-Garcia, J. A., Ruiz, M. D., & Martin-Bautista, M. J. (2024). Exploring social bots: a feature-based approach to improve bot detection in social networks. arXiv preprint arXiv:2411.06626.

[4] Nguyen, H. D., Nguyen, D. Q., Nguyen, C. D., To, P. T., Nguyen, D. H., Nguyen-Gia, H., ... & Quan, T. (2024). Supervised learning models for social bot detection: Literature review and benchmark. Expert Systems with Applications, 238, 122217.

[5] Sallah, A., Agoujil, S., Wani, M. A., Hammad, M., Maleh, Y., & Abd El-Latif, A. A. (2024). Fine-tuned understanding: Enhancing social bot detection with transformer-based classification. IEEE Access, 12, 118250-118269.

[6] Mou, G., & Lee, K. (2020, October). Malicious bot detection in online social networks: arming handcrafted features with deep learning. In International conference on social informatics (pp. 220-236). Cham: Springer International Publishing.

[7] Shevtsov, A., Tzagkarakis, C., Antonakaki, D., & Ioannidis, S. (2022, May). Identification of twitter bots based on an explainable machine learning framework: The US 2020 elections case study. In Proceedings of the international AAAI conference on web and social media (Vol. 16, pp. 956-967).

[8] Deshmukh, A., Moh, M., & Moh, T. S. (2024, December). Bot Detection in Social Media Using GraphSage and BERT. In 2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) (pp. 804-811). IEEE.

[9] Pham, P., Nguyen, L. T., Vo, B., & Yun, U. (2022). Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. Information Systems, 103, 101771.

[10] Javed, D., Jhanjhi, N. Z., & Khan, N. A. (2023, July). Explainable Twitter bot detection model for limited features. In IET Conference Proceedings CP837 (Vol. 2023, No. 11, pp. 476-481). Stevenage, UK: The Institution of Engineering and Technology.

[11] Mbona, I., & Eloff, J. H. (2023). Classifying social media bots as malicious or benign using semi-supervised machine learning. Journal of Cybersecurity, 9(1), tyac015.

[12] Yang, K. C., Varol, O., Hui, P. M., & Menczer, F. (2020, April). Scalable and generalizable social bot detection through data selection. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1096-1103).

[13] Lopez-Joya, S., Diaz-Garcia, J. A., Ruiz, M. D., & Martin-Bautista, M. J. (2025). Dissecting a social bot powered by generative AI: anatomy, new trends and challenges. Social Network Analysis and Mining, 15(1), 7.

[14] Pham, P. (2025). An Integrated Pre-Trained Auto-Encoder and Graph Neural Network for General Social Bot Detection. Annals of Data Science, 1-21.

[15] Terumalasetti, S., & Reeja, S. R. (2024). Enhancing social media user's trust: A comprehensive framework for

detecting malicious profiles using multi-dimensional analytics. IEEE Access.

[16] Paudel, P., Nguyen, T. T., Hatua, A., & Sung, A. H. (2019, August). How the tables have turned: Studying the new wave of social bots on Twitter using complex network analysis techniques. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 501-508).

[17] Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., & Alomari, D. M. (2023). Machine learning-based social media bot detection: a comprehensive literature review. Social Network Analysis and Mining, 13(1), 20.

[18] Wu, J., Ye, X., & Mou, C. (2023). Botshape: A novel social bots detection approach via behavioral patterns. arXiv preprint arXiv:2303.10214.

[19] Bebensee, B., Nazarov, N., & Zhang, B. T. (2021). Leveraging node neighborhoods and egograph topology for better bot detection in social graphs. Social Network Analysis and Mining, 11(1), 10.

[20] Dimitriadis, I., Dialektakis, G., & Vakali, A. (2024). Caleb: a conditional adversarial learning framework to enhance bot detection. Data & Knowledge Engineering, 149, 102245.

[21] Guo, S., Wang, J., Wang, Z., Yu, G., & Wu, S. (2025). BotICC: enhancing social bot detection through implicit connection computation. Multimedia Systems, 31(3), 210.

[22] Qiao, B., Li, K., Zhou, W., Li, S., Lu, Q., & Hu, S. (2025, April). Identifying Bots on Social Media through Coordinated Group Perception. In ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE.

[23] Moghaddam, S. H., & Abbaspour, M. (2022). Friendship preference: Scalable and robust category of features for social bot detection. IEEE Transactions on Dependable and Secure Computing, 20(2), 1516-1528.

[24] Ellaky, Z., Benabbou, F., Ouahabi, S., & Sael, N. (2021, June). A survey of spam bots detection in online social networks. In 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA) (pp. 58-65). IEEE.

[25] Wu, J., Ye, X., & Man, Y. (2023, May). Bottrinet: A unified and efficient embedding for social bots detection via metric learning. In 2023 11th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

[26] Najari, S., Salehi, M., & Farahbakhsh, R. (2022). GANBOT: a GAN-based framework for social bot detection. Social Network Analysis and Mining, 12(1), 4.

[27] Sánchez-Corcuera, R., Zubiaga, A., & Almeida, A. (2024). Early Detection and Prevention of Malicious User Behavior on Twitter Using Deep Learning Techniques. IEEE Transactions on Computational Social Systems.

[28] Kouvela, M., Dimitriadis, I., & Vakali, A. (2020, November). Bot-Detective: An explainable Twitter bot detection service with crowdsourcing functionalities. In Proceedings of the 12th International Conference on Management of Digital EcoSystems (pp. 55-63).

[29] Li, Y., Li, Z., Gong, D., Hu, Q., & Lu, H. (2024). BotCL: a social bot detection model based on graph contrastive learning. Knowledge and Information Systems, 66(9), 5185-5202.

Kumar, A. S., Kumar, N. S., Devi, R. K., & Muthukannan, M. (2023). Analysis of deep learning-based approaches for spam bots and cyberbullying detection in online social networks. AI-Centric Modeling and Analytics, 324-361.