Utilizing Machine Learning for Classification of Tweets as Human or Bot Generated

Adrian Nugroho Basuki
Computer Science Department
School of Computing and Creative Arts
Bina Nusantara University,
Jakarta, Indonesia 11480
adrian.basuki@binus.ac.id

Fadhillah Haidar Rahyang
Computer Science Department
School of Computing and Creative Arts
Bina Nusantara University,
Jakarta, Indonesia 11480
fadhillah.rahyang@binus.ac.id

Kevin Jonathan
Computer Science Department
School of Computing and Creative Arts
Bina Nusantara University,
Jakarta, Indonesia 11480
kevin.jonathan014@binus.ac.id

Nunung Nurul Qomariyah
Computer Science Department
School of Computing and Creative Arts
Bina Nusantara University
Jakarta, Indonesia 11480
nunung.qomariyah@binus.edu

Raymond Bahana
Computer Science Department
School of Computing and Creative Arts
Bina Nusantara University
Jakarta, Indonesia 11480
rbahana@binus.edu

***Abstract*—In the modern world, the Internet has become a place for communicative purposes, such as spreading information to a large group of users through social media platforms like Twitter. A major problem with social media is the misuse of bots. A bot is a created program that allows developers to assign certain tasks to automate human activities in an indefinite time. Based on the current popularity with social media platforms, countless bots are abused to spam similar messages to gain exposure; the exposed material can include phishing, gambling promotions, etc. Conducted research is necessary to study and analyze these activities in hopes of reducing redundant tweets and illicit content. This research involved acquiring sufficient tweets, applying feature engineering, and feeding the processed dataset to six classification machine learning models for predictions. It was revealed that, generally, bot tweets are more likely to contain links. The reason for this is because bots in social media are often used to invite people to click on links, whether that would be to scam the audience or to promote their business. Additionally, Random Forest, XGBoost, and K-Nearest Neighbors were concluded to have the best performances with accuracies of 94.61%, 94.77%, and 93.72% respectively. An explanation to this result is that these machine learning models are able to handle unbalanced datasets.**

***Index Terms*—Machine learning, Classification, Predictions, Social media, Bot detection**

I. Introduction

In the current era of the Internet, social networks have played an important role in entertaining people, spreading news, and promoting businesses to share the content of the business or increase sales [1]. One of the ways to do this is by having assigned personnel to manually create posts or advertisements and upload them on a social media platform such as X, formerly known as Twitter. Despite its simplicity, it can be inefficient due to certain factors, namely, human error, availability, health conditions, etc. All these problems can be solved by automating the messages or posts using a bot. However, as previously mentioned, this solution has several drawbacks, such as abusing it to spread hate by sharing messages with negative sentiment or messages containing hashtags with negative connotations [2]. This misuse reduces the trust in the use of bot accounts, as it could cause further problems, such as manipulation of public opinion, a higher likelihood of getting exposed to explicit content, and making it easier to trick people into accessing malicious links.

979-8-3315-6791-0/25/$31.00 © 2025 IEEE

Due to this ongoing problem, our goal is to train several chosen machine learning models to find the model that provides the best accuracy for predicting tweets made by a bot. The current problem is that, even though users have the ability to block or mute bot accounts on social media platforms, it is impossible to prevent bot tweets from appearing, as new bot accounts keep being created. Hence, it is recommended to minimize their exposure to online users instead. This solution can be beneficial to the developers of X, as well as users on the platform. For the developers, removing some of these tweets can provide extra space in their database. This becomes an excellent investment where storage becomes largely in demand in the future. In the case of normal users, having lower counts of bot tweets leads to less spam in their page and reduces chance of encountering harmful or deceptive content produced by bots. This helps create a safer environment when browsing through social media platforms.

To ensure the effectiveness of our selected machine learning models, we trained the models with relatively new tweets, then made hypotheses and questions. Hypothetically, analyzing the direct features from a number of tweets within a given time frame can be useful to train a program in detecting certain patterns. In addition, it can be hypothesised that additional features based on existing features could further improve the detection and be used for future reference to detect newer bot tweets. A question can be made for which features can effectively contribute to determining a bot tweet? A follow up question is, can our scraped data be used to detect all types of bots? Finally, the dataset we created includes several columns of data for tweets created between January to September, 2024.

Lastly, conducting research involving self-scraped data for bot tweet detection has its limitations. At the start of 2023, Twitter stopped offering free access to their API. Therefore, we used a tool called TweetHarvest to crawl data, the tweets, from our homepages [3]. Although TweetHarvest also has some rate-limitation problems, it still allows us to scrape decent amounts of data within a short time when used correctly. Other than rate limitations, another limitation would be providing labels to the harvested tweets. An existing Twitter bot account detector, Botometer, is limited to checking Twitter accounts made before June 2023 [4]. Since we focus on tweets created beyond 2024, we could not rely on this tool to help us classify whether the tweet creator is a bot account or not. Therefore, we classified them ourselves based on the similarity with other tweets, as bots tend to create tweets with similar content.

II. Related Work

This section will discuss a few research papers related to our topic. These are some papers we deemed related.

First is a research paper on machine learning - based social media bot detection. This research paper is a literature review that compiles and reviews research papers on bot detection on social networks from 2015 to 2022 to find out which social networks are the most commonly used for research and which machine learning model works best on bot detection. Based on the findings of the paper, Twitter was the most commonly used social media for research. This is due to the amount of datasets available online and how easily accessible they are to be used to scrape data. Despite the amount of available datasets on the Internet, most datasets lack information or attributes for the accounts, which makes most researchers gather their own datasets instead using APIs or libraries that exist, such as the Twitter API. In the context of Twitter platform, the most common type of features used for training the machine are profile-based feature and content-based. Profile-based features are characteristics of a user's profile, such as the user's profile picture, bio, age of the account, etc. Content-based features are coming from the content that the users post on social media. For example, it could be like word frequencies and topics within the content. The most frequently used and best performance machine learning model for bot detection is random forest, and followed by support vector machines [5].

Other than that, there is another paper from 2023 that highlights the use of Deep Learning techniques for social bot detection. Deep learning is a subset of machine learning involving more complex models than traditional machine learning. The paper provides a systematic review of 40 selected publications published between 2000 and 2021 about various Deep Learning approaches, mainly for bot detection in social media. Other than that, they have also compared the effectiveness of different Deep Learning algorithms, as well as compared them to Machine Learning approaches for bot detection. This paper showed that deep learning approaches perform better or as well as traditional machine learning languages, reaching accuracies up to 100%. However, to achieve such accuracy, the model needs to be trained with a very large dataset from hundreds of thousands to millions of data samples [6].

Lastly, there is a 2024 research paper that specifically explores the detection of political social media bots using machine learning. This paper aims to review existing studies on the detection of political bots on social media and address challenges, such as feature engineering and feature selection, that may affect the accuracies of the bot detection models. They have extracted over 33 features, including content and user information, from the Twibot-20 dataset, a public dataset containing data for Twitter bots and genuine users in 2020 and 2021. They explored several feature selection techniques to select an array of the most optimal features which would then be used to train multiple machine learning algorithms. After processing their datasets, using techniques such as synthetic minority oversampling technique to balance their datasets, the results showed a significant improvement in their results for AUC and accuracy scores. At the end of their research, they concluded that Random Forest, Decision Trees, Adaboost, XGBoost, Gradient Boosting, and Extra Trees are the most effective machine learning models for detecting political bots in social media [6].

Previous studies have provided valuable insights and act as a baseline for our research. Even though research shows that deep learning can perform better for bot detection, we decided to use traditional machine learning because we lacked the time and tools required to gather hundreds of thousands to millions of recent sample data. Traditional machine learning is more suitable with smaller datasets, while deep learning models will perform worse. Different from the related works mentioned above, our research aims to detect a broader range of bots using an updated dataset.

III. Methodology

A. *Dataset Overview and Description*

For our dataset, we used the Tweet Harvest tool to retrieve the data from a famous social platform, X or formerly known as Twitter. This simulates a human scrolling through the homepage of X to minimize getting rate-limited. From this we were able to produce raw data consisting of 15 columns, which includes conversation ID, the time of tweet creation, number of favorites of the tweet, the full text, the tweet ID, any present image URLs, checking if it is a reply, the language, location, number of quotes, number of replies, number of retweets, the tweet link, user ID, and the username of the creator of the tweet. We managed to retrieve more than 16000 rows of data using certain keywords that guarantee bot and human tweets and crawled data from January to September 2024.

Our method of data collection involves utilizing common keywords that may be used by bots or humans. To ensure

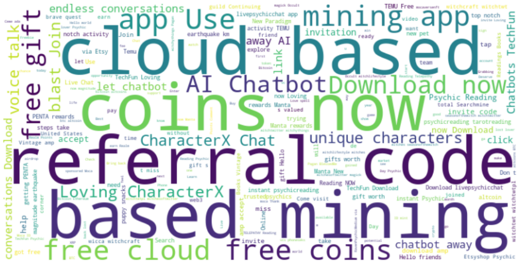

Fig. 1. Word cloud of bot-generated tweets.

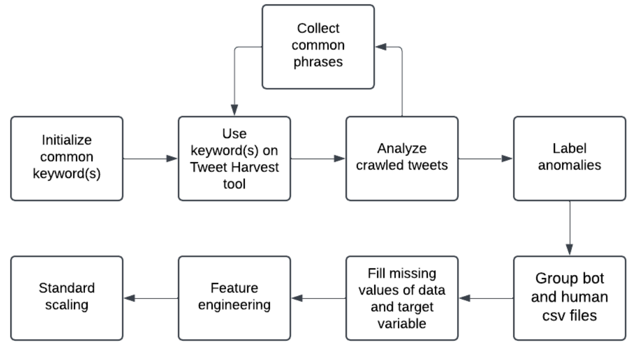

Fig. 2. Data collection and preprocessing flowchart.

efficiency, the keywords used should give either all bots, or all human tweets. With our prior knowledge of certain phrases that are commonly used by bots, we used those as keywords to get some bots. One of these keywords is ‘Click this link’. From there, we analyze each tweet and check for similar tweets having the same message. An example phrase derived from this method is ‘cloud based mining app’. After gathering a number of keywords deemed to be bots, we use those keywords to find more similar tweets in the set time frame. Figure 1 illustrates common word phrases from bot tweets in a form of a word cloud.

B. Dataset Preprocessing

Figure 2 shows the flow of preprocessing the data. The initial step was to use common word phrases to search possible bot tweets, such as ’click this link’. Then, manual analysis is done towards the generated csv file to obtain phrase duplicates to be later used in searching for more bots. If a csv file consist of similar tweets, manual anomaly labelling is done on supposed bot and human csv files. In order to do this efficiently, for every csv file that uses a keyword to get bot tweets, manually label the human tweets. For every human csv file, manually label the bot tweets. After manual labelling, we combine the csv files for bots and humans. Once combined, every missing value in the label column is labeled as human for the human dataset, and bot for the bot dataset, completing the labelling step. For handling missing values in some columns, we either input ‘None’ or ‘unknown’ based on the attribute’s description. This is because a few attributes, such as image_url, will be checked to see if the tweet contains them or not, which will be useful in looking for patterns of tweets made by humans and bots.

Once cleaned, we add additional features derived from the existing features to be used later in machine learning. These features involve with boolean values and numerical lengths, such as checking for links, checking media attachments, the tweet length, the hashtag count, the punctuation count, and the capital count. Lastly, standard scaling, using scikit-learn’s standard scaler, is done as some machine learning algorithms are affected by standardisation. Table I shows the raw dataset collected from the tweet harvest tool. Table II shows the dataset after preprocessing and cleaning.

C. Model and Techniques

In this research, we trained several machine learning models with the data that we scraped. These machine learning models include Random Forest, Decision Tree, XGBoost, SVM, Logistic -Regression, and K-Nearest Neighbors.

1) Random Forest: A classifier machine learning model developed by Leo Breiman that consists of a group of tree-structured classifiers, such as decision trees [7].
2) Decision Tree: The Decision Tree model is a machine learning model that consists of nodes that form a directed tree with a root node with no incoming edges. It divides the data based on attribute values and could be used for prediction by following each path from the root of a decision tree to one of its leaves [8].
3) XGBoost: A scalable machine learning system utilized for tree boosting, which involves combining weak decision trees sequentially to then become a strong one [9].
4) Support Vector Machine: SVM is a machine learning model that forms a suitable decision boundary, the hyperplane, to separate n-dimensional space into multiple classes [10]. For our case it would be 2 dimensional space.
5) Logistic Regression: A machine learning model where the response variable is binary. A linear combination of independent variables and the probability of a specific event constructs the relationship of this model, utilizing the logit link function [11].
6) K-Nearest Neighbors: K-Nearest Neighbors utilizes distance and set classes to continuously group data into the classes with continuous improvement of placement. Each class is assigned, in the feature space, to the label of the majority K-nearest data points [12].

D. Evaluation Techniques

To find out the best machine learning model for detecting bots, we need to assess the performances of each model. We implemented a variety of evaluation techniques, including precision, recall, f1-score, accuracy, macro and weighted average, and confusion matrix. These techniques were chosen to make

TABLE I
TWEET INFORMATION DATA SET

Attribute	Description	Sample Data
created_at	The time of tweet creation	Thu Feb 01 23:59:51 +0000 2024
favorite_count	The number of likes of a tweet	2
full_tweet	The full message of the tweet	@mainmajin Bro really tried to sneak Asa https://t.co/NNHheuV6ji
tweet_id	The tweet ID	1753206558995980000
image_link	The link of an image from a tweet	https://pbs.twimg.com/media/GFSkEpdXQAAimRl.jpg
in_reply_to_tweet_name	The user that the tweet is replied to	mainmajin
tweet_language	The language of the tweet	en
location	The location of the user, set by themselves	Texas, USA
quote_share_count	The amount of times the tweet has been shared as a quote	0
reply_count	The amount of replies a tweet received	2
retweet_count	The amount of times the tweet has been retweeted	0
tweet_link	The link to the tweet	https://x.com/Samuel3k666/status/1753206558995984793
user_id	The ID that represent the person that posted the tweet	1675025783461400000
username	The name of the account that posted the tweet	Samuel3k666
account_type	The target variable, classifying as bot or human	human

TABLE II
PROCESSED TWEET INFORMATION DATA SET

Attribute	Description	Sample Data
created_at	The time of tweet creation	Thu Feb 01 23:59:51 +0000 2024
favorite_count	The number of likes of a tweet	2
full_tweet	The full message of the tweet	@mainmajin Bro really tried to sneak Asa https://t.co/NNHheuV6ji
tweet_id	The tweet ID	1753206558995980000
image_link	The link of an image from a tweet	https://pbs.twimg.com/media/GFSkEpdXQAAimRl.jpg
in_reply_to_tweet_name	The user that the tweet is replied to	mainmajin
tweet_language	The language of the tweet	en
location	The location of the user, set by themselves	Texas, USA
quote_share_count	The amount of times the tweet has been shared as a quote	0
reply_count	The amount of replies a tweet received	2
retweet_count	The amount of times the tweet has been retweeted	0
tweet_link	The link to the tweet	https://x.com/Samuel3k666/status/1753206558995984793
user_id	The ID that represent the person that posted the tweet	1675025783461400000
username	The name of the account that posted the tweet	Samuel3k666
account_type	The target variable, classifying as bot or human	human
text_has_link	Checks if the text has a link	True
has_media_attachment	Checks if the tweet has an attached image	True
tweet_length	The length of the tweet message	64
hashtag_count	The hashtag count in a tweet	0
punctuation_count	The number of punctuations used	1
capital_count	The number of capital letters	6

sure that these machine learning models will have a good reliability and accuracy in distinguishing between human and bot accounts.

1) Accuracy: Accuracy is the most used evaluation technique used to measure the performance metric of a machine learning model. The ratio of the total number of correct predictions to the total number of predictions is measured. The drawback of accuracy is when it is used in measuring the accuracy of a machine learning model that is working on imbalanced datasets. If a model classifies that the majority of the large dataset belongs to the major class label, it will perform poorly in classifying the minority class label, even though the model has a high accuracy [13].
2) Precision: Precision only analyses the positive outcomes, it measures the ratio of true positive outcomes to the sum of true positives and false positive outcomes. Using precision alone to measure the performance of the machine learning algorithm is not sufficient because it does not take the negative outcomes into account [14].
3) Recall: Recall analyses the correct number of correct positive outcomes. It measures the ratio of true positives and the sum of true positive and false negative outcomes. The drawback of recall is that it often leads to a

higher false positive rate because it only penalizes false positives, but not false negatives [15].

4) F1 Score: This evaluation metric uses both recall and precision values for its measurement, also known as the harmonic mean. It is good for evaluating a machine learning model if there happens to be trade-offs between precision and recall, and you want to find the balance between the two measures, since it considers both metrics equally. This makes F1 score very useful in scenarios where there is an imbalance of datasets where focusing on either recall or precision only could be inaccurate [16].
5) Weighted Average: Weighted average in terms of metric calculation, including precision, accuracy, F1-score, and recall, refers to an approach where we consider the contribution of each class based on its support. A support refers to the number of true occurrences in each class. The metric of each class is multiplied by the proportion of support in that class, and then all these values are summed to get the overall metric [15].
6) Macro Average: The macro average in terms of metric calculation, is measured by calculating the unweighted mean of all per-class metrics. It means that each class is treated equally, regardless of the proportion of its support [17].
7) Confusion Matrix: A confusion matrix is a N x N matrix where N is the amount of target classes. It is used to describe the performance of machine learning models, primarily classification models, by showing the number of predicted values that are correct, which are true positives and true negatives, and the number of predicted values that are incorrect, which are false positives, and false negatives. It is a good tool to use to calculate the other metrics mentioned previously and it can help to identify specific areas where the machine learning model is making errors [18].

IV. Results and Discussion

After training our selected machine learning models, we used the test subset to check our models' performances. Our chosen evaluation metrics are then used on the test subset's prediction results to assess the performance of each model with more detail. Based on the results we received, we found that Random Forest, XGBoost, and K-Nearest Neighbors are the three most effective models for our dataset. This is because they have the highest accuracies and f1-scores compared to the other machine learning models, as well as having the least numbers of false positives and negatives. Other than that, the precision, recall, and f1-scores for these three machine learning models also show the smallest differences between their calculations for bot and human classes. These results show that those machine learning models are capable of handling our unbalanced dataset that has more human data compared to bot data.

We have also analyzed the feature importance based on some of our tree-based machine learning models. The feature importance values are measured using the mean decrease in impurity (MDI), which computes the reduction in impurity contributed by all splits for a certain feature in a decision tree [19]. A higher MDI value shows that the feature contributes more in improving the decision-making process of the machine learning model.

TABLE III
Classification Report for Selected Models

Model	account_type	Precision	Recall	F1-Score	Accuracy
SVM	Human	0.95	0.94	0.95	0.9255
	Bot	0.87	0.89	0.88	
XGB	Human	0.96	0.97	0.96	0.9477
	Bot	0.92	0.90	0.91	
DT	Human	0.95	0.93	0.94	0.9119
	Bot	0.83	0.88	0.86	
LR	Human	0.94	0.93	0.94	0.9089
	Bot	0.85	0.85	0.85	
KNN	Human	0.96	0.95	0.96	0.9372
	Bot	0.89	0.90	0.89	
RF	Human	0.96	0.96	0.96	0.9461
	Bot	0.91	0.91	0.91	

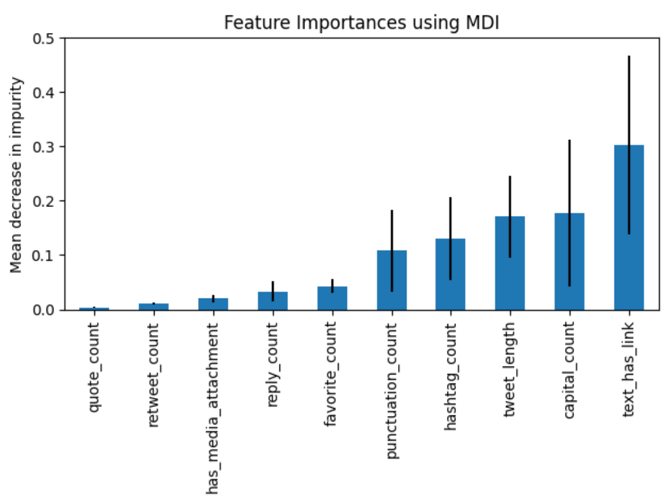

Fig. 3. Feature importance for Random Forest model.

From our feature importance analysis, we found that the 'text_has_link' feature has the highest MDI value for all the models we tested. After conducting further analysis, we have also proven that the percentage of bots that have links in their tweets is much higher compared to humans. This means that bots are more likely to include links in their tweets, which is because our dataset contains a lot of promotional or spam-like bots that tell the users to click links in their tweets to access the sites they want to promote.

Aside from 'text_has_link,' other features, such as 'tweet_length,' 'hashtag_count', 'punctuation_count,' and 'capital_count,' also contribute to the decision, but to a lesser extent. This may be because these features show some difference between bot and human behaviors, but are not as significant compared to the presence of links in the tweet.

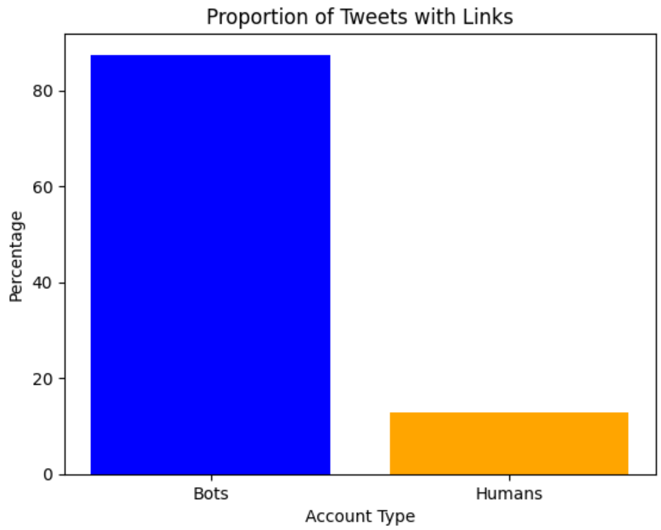

Fig. 4. Proportion of bot and human accounts with links in their tweets.

V. Conclusion and Future Work

To conclude our research, we have found that the three most effective machine learning models for our dataset are RandomForest, XGBoost, and K-Nearest Neighbors. This is because they have the three highest accuracies among all the models we have experimented with, and have the highest precision, recall, and f1-scores with small differences between bot and human classes, showing that those models are the best at handling our unbalanced dataset.

Other than that, we have also explored the importance of features for some of our tree-based machine learning models, such as XGBoost and Decision Tree. Based on the results, we have concluded that 'text_has_link' is the feature that contributes the most in deciding whether the tweet was written by a bot or human. Some other features, such as 'hashtag_count', also contribute to the decision, but not as much as the 'text_has_link' feature.

For the future of this research, one option we could go with is to explore how we could detect bots in social media platforms that utilize Language Learning Models (LLM). With the invention of various Language Learning Models, such as ChatGPT, people worldwide have been using them for many different purposes in their lives. However, people could misuse them in many ways, including using them to make dangerous social media bots appear human.

Another option for our future work is to discover the topics that bot accounts frequently discuss. By identifying the topics that bot accounts discuss, we could find out the bot's objectives and the types of content they want to share. Additionally, this information could be useful in detecting harmful bot behavior.

Supplementary Codes

All the codes used in this paper can be accessed through the following link: https://github.com/AdrianBasukii

Credit Authorship Contribution Statement

Adrian Nugroho Basuki, Fadhillah Haidar Rahyang, Kevin Jonathan: Conceptualization, Data Curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Nunung Nurul Qomariyah, Raymond Bahana**: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

[1] W. Akram, "A study on positive and negative effects of social media on society," *International Journal of Computer Sciences and Engineering*, vol. 5, no. 10, pp. 351–354, Oct. 2017.

[2] M. Stella, E. Ferrara, and M. D. Domenico, "Bots increase exposure to negative and inflammatory content in online social systems," *Proceedings of the National Academy of Sciences*, vol. 115, no. 49, pp. 12 435–12 440, Nov. 2018.

[3] H. Satria, "Crawl data twitter menggunakan tweet harvest - juli 2023," https://helmisatria.com/blog/crawl-data-twitter-menggunakan-tweet-harvest/, accessed: Dec. 15, 2024.

[4] Botometer X, "Botometer x," https://botometer.osome.iu.edu/, accessed: Dec. 15, 2024.

[5] M. Aljabri *et al.*, "Machine learning-based social media bot detection: A comprehensive literature review," *Social Network Analysis and Mining*, vol. 13, no. 1, 2023.

[6] 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, Sep. 2024.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] L. Rokach and O. Maimon, "Decision trees," in *Data Mining and Knowledge Discovery Handbook*, 2005, pp. 165–192.

[9] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

[10] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, p. 100071, 2022.

[11] N. Srimaneekarn, A. Hayter, W. Liu, and C. Tantipoj, "Binary response analysis using logistic regression in dentistry," *International Journal of Dentistry*, vol. 2022, no. 1, p. 5358602, 2022.

[12] O. Kramer and O. Kramer, *K-Nearest Neighbors. Dimensionality Reduction with Unsupervised Nearest Neighbors*, 2013.

[13] C. Sweeney, E. Ennis, M. Mulvenna, R. Bond, and S. O'Neill, "How machine learning classification accuracy changes in a happiness dataset with different demographic groups," *Computers*, vol. 11, no. 5, p. 83, 2022.

[14] H. Dalianis, "Evaluation metrics and evaluation," in *Clinical Text Mining*, 2018, pp. 45–53.

[15] G. Varoquaux and O. Colliot, "Evaluating machine learning models and their diagnostic value," in *Neuromethods*, 2023, pp. 601–630.

[16] H. Kaur and D. N. K. Sandhu, "Evaluating the effectiveness of the proposed system using f1 score, recall, accuracy, precision and loss metrics compared to prior techniques," *Int. j. commun. netw. inf. secur.*, vol. 15, no. 4, pp. 368–383, 2024.

[17] D. K. Gichuki, P. O. Owoche, and S. M. Mbuguah, "Performance evaluation of machine learning algorithms in smart agriculture," *IJARCCE*, vol. 13, no. 8, 2024.

[18] S. Sathyanarayanan, "Confusion matrix-based performance evaluation metrics," *African Journal of Biomedical Research*, pp. 4023–4031, 2024.

[19] X. Li, Y. Wang, S. Basu, K. Kumbier, and B. Yu, "A debiased mdi feature importance measure for random forests," *ArXiv*, vol. abs/1906.10845, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:195658234