University of Ibadan
Journal of Science and Logics in ICT Research

Comparative Analysis of Machine Learning Algorithms for the Classification of Twitter Bots

[1]✉Ayogu, B. A., [2]Ogunleye, G. O., [3]Adewole, L. B., [4]Olagunju, M. and [5]Oyatoyinbo, W. A

Department of Computer Science, Federal University, Oye Ekiti, Nigeria.
bosede.ayogu@fuoye.edu.ng,gabriel.ogunleye@fuoye.edu.ng, lawrence.adewole@fuoye.edu.ng, mukaila.olagunju@fuoye.edu.ng, oyatoyinbowunmi@gmail.com

Abstract

Social media platforms have become risky for actual users due to the rise in the number of bots. The security mechanisms put in place to help identify and categorize bots accounts from legitimate human accounts have significant drawbacks, such as the misclassification of accounts because of behavioral change. In general, studies on Twitter bots identification demonstrate that bots can be useful while also having a negative impact on users by broadcasting misleading news, spamming, or posing as a phony follower to boost an account's popularity. This study employed Logistic Regression, Catboost, and Random Forest algorithms to develop Twitter bots classification systems, capable of distinguishing between useful and harmful bots accounts in order to limit their impact on users and the Twitter community. The feasibility of the algorithms was tested on Twitter spam bots dataset gotten from Kaggle, containing eight(8) features, which were reduced to two (2) using decision tree. The selected features were further utilized to develop bots classification systems. Comparative analysis of the results showed that Random forest classifier recorded best performance when evaluated on training set, while the Logistic recorded highest performance in terms of accuracy, precision, and F1 Score achieving 83%, 78%, and 81%, respectively when evaluated on test set. The classification systems can help identify and mitigate the impact of harmful bots on Twitter, such as those used for spamming or disseminating fake news. The study has demonstrated the effectiveness of machine learning algorithms in classifying Twitter bots and provided a potential solution for improving online social media platforms.

Keywords*: Catboost; Decision tree; Feature Selection; Logistic Regression; Random Forest*

1. Introduction

Twitter, now X was established to be a social networking site for micro blogging that allows registered users to post and read 140-character messages called tweets [1]. Unlike on MySpace or Facebook, the relationship between following accounts and being followed is not needed on Twitter. This means an account can be followed by other accounts without following them back. The recognition and influence of twitter on modern society has generated a huge amount of multimedia information that are rapidly disseminated on the platform[2].The rapid usage of the Twitter by different individuals has made it to be vulnerable, quite a number of important parts of the accounts in this social network are not humans. As seen in the literature, 15% of all active registered twitter users are bots [3].

Bots are automated programs that manage Twitter accounts. Twitter bots, or automated Twitter programs, are becoming increasingly prevalent as Twitter grows in popularity. Nevertheless, studies on Twitter bots identification demonstrate that bots can be useful while also having a negative impact on users by broadcasting misleading news, spamming, or posing as a phony follower to boost an account's popularity [4]**.** Twitter bots can potentially alter public perceptions of an issue, reduce user confidence, and even disrupt social order. Quite a large volume of tweets has the potential to pollute a user's timeline, change their perception, damage their confidence, disrupt the stock market, and even undermine societal order. With all of these challenges, distinguishing between legitimate

Ayogu, B. A., Ogunleye, G. O., Adewole, L. B., Olagunju, M. and Oyatoyinbo, W. A (2024). Comparative Analysis of Machine Learning Algorithms for the Classification of Twitter Bots, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 11 No. 1, pp. 63 - 73

user accounts and bots accounts is extremely tough [5].

Identifying bots accounts and non-bots accounts has traditionally been done by looking at an account's activity pattern, for example, observing that a certain user re-tweets more than it creates its own tweets, tweets simultaneously, and has just few followers. Furthermore, the tweeting account lacks a biography and a profile picture, and it tweets identical text as another user account simultaneously. On the other hand, such traditional/cognitive approaches are deemed wasteful because they only focus on precision. An alternative to the conventional way of differentiating bots from non-bots is the use of machine learning (ML), which is an application of Artificial intelligence (AI) that aids in giving systems the capacity to learn on their own, as well as improving the experience of the programmed system. Different measures have been applied by various researchers to detect and remove bots from social media networks [6] [7][8] [9]. Aljabri *et. al.*[10] did a comprehensive review of recent advancements in machine learning-based techniques for the detection and classification of bots on five major social media platforms, namely Facebook, Instagram, LinkedIn, Twitter, and Weibo.

Despite the various ways and techniques used in Twitter bots classification, misclassification still occurs on various users' accounts, i.e., classifying real user accounts as bots due to behavioral change, location, user identity(ID), and the rate at which user responds to tweets, etc. Nevertheless, some bots are still used to share useful information. This research aims to classify bots into harmful and not harmful while focusing on spamming bots. The remaining parts of this paper are structured into four sections. Section two gives a detailed report of related works. In section three, the methods and frameworks used in achieving the main objectives of this research are described. An experimental analysis of the results obtained from the developed systems is also presented in section four. Finally, conclusion and directions for future research are given in section five.

2. Related Works

There are lots of researches on social bots detection many of which applied machine leaning/deep learning even though they all tried solving same problem with different approaches. Liu *et. al.* [11] proposed a social bots classification system where polluter was identified by using topic related introduction to expand the topic. Sentence-Bert model was applied to make relevance judgments between the micro-blog text and the expanded. To further differentiate commenters and spreaders, an opinion sentence recognition method that combines social bots' opinion sentence generation rules with a deep learning model (Text CNN) was also proposed.

Jalal and Ghafoor [1] initiated a new benchmark created on a 1.5m Twitter profile. In detecting Twitter bots, different supervised machine-learning models were trained on the dataset. In the case of generalization, their benchmark achieved a 6% higher area under the curve than another dataset when the models were trained with other state-of-the-art benchmarks.

Alarfaj [4] applied five different classification algorithms to detect Twitter bots. To get real-world from Twitter, Twitter API was used and further pre-processed using the min-max normalization technique; features were selected using Information Gain, Gain Ratio, and Relief- F. Performance evaluation was done by testing data with and without normalization where the dataset without normalization achieved a very low result.

Khaled [12] investigated the detection of fake (Sybil) Twitter accounts using a hybrid supervised machine learning model of SVM+NN, realizing accuracy of 98%. Wei, and Nguyen[13] employed bidirectional long short-term (BiLSTM) to capture features across tweet. The model was used with word embedding to distinguish Twitter bots from human accounts. The experiments were carried out on the Cresci-2017 dataset to validate their model, and the results were compared with the existing state-of-the-art bot detection system.

Nguyen*et al.* [14] developed a topological feature extraction for bots detection on social media networks. A weighted ego network of

each user was created; thereafter, the higher-order topological features of the ego network were encoded using persistent homology. The extracted features were further used to train machine learning model for classification purpose. Ramalingaiah *et. al.* [3] applied decision tree, k-nearest neighbor, Logistic Regression, and Naive Bayes to the detection of Twitter bots. Efficacies of the models were demonstrated on Twitter bots dataset by comparing them with other classifiers that uses Bag of bots' word model. The results of the evaluation recorded SVM to achieve the best performance with a recall of 89% and F1 score of 76%.

Mbona, and Eloff [15] employed various machine learning techniques to design systems that could classify bots into malicious and benign. In their work, features that indicated anomalous behavior between benign and malicious bots were identified. Among the supervised machine learning techniques employed, the support vector machine achieved the highest performance in classifying malicious and benign bots when evaluated on the Twitter dataset. The performance of their models cannot be guaranteed as the experiment was validated on a limited dataset. Nevertheless, a fundamental understanding of the differences between harmful bots and non-harmful bots accounts is essential. On this note, this research employs three different machine learning algorithms (CatBoost Algorithm, Random Forest Classifier, and Logistic Regression) to the detection and classification of harmful and non-harmful Twitter bots to limit their impact on users and the Twitter community and also compare the results of the models based on the standard performance evaluation metric.

3. Methodology

This section comprises of the methods used in achieving the proposed system, divided into various parts. Three different machine learning models are employed and implemented independently to classify Twitter bots into harmful and non-harmful and make a comparative analysis of their performance using standard performance evaluation metrics. The models employed are CatBoost, Random Forest, and Logistic Regression. Figure1 is the architectural design of the proposed system, comprising of Dataset Collection, Dataset Cleaning, Data Splitting, Modeling, and classification.

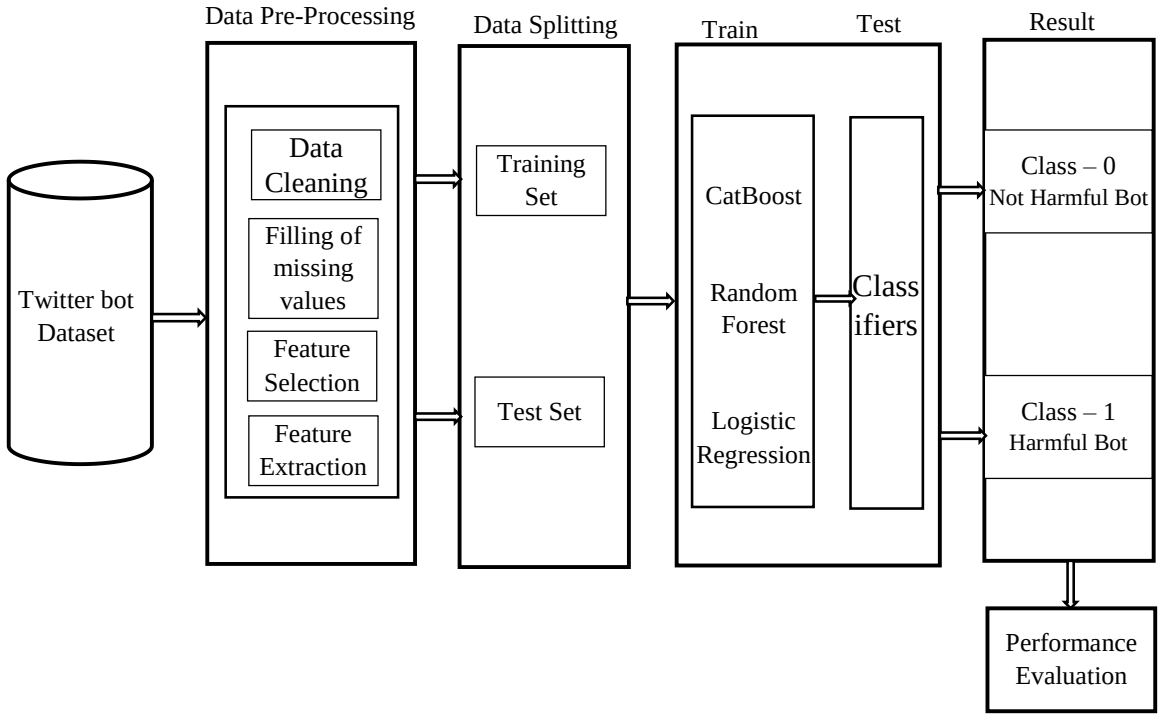

Figure 1: Architecture of the proposed system.

Table 1: Dataset Attributes with their interpretation.

S/N	Features	Feature Interpretation
1	id	They are unique integers used in identifying individual/tweeting account.
2	Tweet	The tweeted Text
3	Following	Number of registered users the tweeting account follows
4	Followers	Number of registered users that follows the account that tweeted
5	Actions	Total number of favourites, replies, and retweets of a particular tweet
6	Is_retweet	[0, 1] Binary value: If 0, it is not a retweet; if 1, it is retweet
7	Location	Self-written location provided by registered user on their profile, May not exist, be "Unknown", and is NOT standardized! An example could be ("New York", "N. Y.", "Upper East Slide", etc.)
8	Type	0 or 1 (0s for a not-harmful bot or 1s for a harmful bot)

3.1 Data collection and preprocessing

For the purpose of this research, the Twitter bots dataset was obtained from https://www.kaggle.com/c/twitter-spam/data, with the extension of CSV consisting of 11968 instances split into 80% and 20% training and testing sets, respectively. There are 7 conditional and 1 decision attributes represented in the dataset with three datatypes; int64(1), float64(5), and object(2) as represented in Table 1.The harmful instances in the dataset are represented by 1, which contains 5815 records, while the non-harmful instances are represented by 0, which contains 6153. The dataset sample is presented in Table 2.

Data cleansing, or eliminating/substituting values for invalid, empty, or Not Number (NaN) values, is the first step in the pre-processing process, followed by deletion of duplicate data if any is found. Determining the used parameters as a set analysis feature comes next. The information structure was altered through processing so that the Machine Learning classifier algorithms could process it. Only numerical values can be processed by the method.

Table 2: Initial Dataset

INSTANCES	
Not harmful (0)	6153
Harmful (1)	5815
TOTAL	11968

3.2 Experimental Procedure

To prepare the data sets for suitability of the algorithms employed in this research, there was a need for data organization and cleaning, involving the removal of duplicate or irrelevant observations, handling missing values, and fixing of structural errors. To fill in the blank rows in the dataset, the median of a numerical variable was used, and a forward-fill approach for categorical variables was used to fill the missing rows as shown in Equation 1.

$$\text{Median} = I + \left[\frac{\frac{m}{2} - d}{e}\right] \times g \qquad (1)$$

This procedure was carried out using Python libraries Pandas. $Fillna()$function was used to fill all the missing rows. For the duplicated values, Pandas' *duplicated()* method only looks at duplicated values. It only returns a series of Booleans that are True if the entry is Unique. To determine the feature importance of individual feature in the dataset, Information Gain was considered. First, the entropy of the data was calculated as shown in Equation 2.

$$Entropy = \sum_{i=1}^{d} -E_j \log_2 E_j \qquad (2)$$

Where d is the quality of unique class labels, and E_j is the proportion of instances with output labels in j.

There was also a need for feature extraction, for the purpose of this research, the features were extracted from the terms returned from the

preprocessing using Term frequency - inverse document frequency $(TF-IDF)$Vectorizer.To translate the text into a numerical representation that can be used as input into a machine learning method, the terms obtained from the preprocessing must be encoded as integers or floating-point numbers. The news headlines (text) were turned into vectors. The relevance of each keyword in the collection of the documents was determined using the IDF, which is the inverse of the D.F. IDF is described in Equation 3.

$$IDF_{j,l} = \log \frac{|D|}{|dl \in D : t_l \in d_l|} \quad (3)$$

When a certain term frequently appears in a text but inadequately frequently across the full corpus of documents, the $TF-IDF$ score rises. Utilizing this concept, the terms that frequently appear in publications can be located. Following the feature extraction, there is a crucial process known as classification, the purpose of which is to categorize the unseen news into the appropriate groups. this study utilized three different machine learning algorithms (CatBoost Logistic Regression, and Random forest) to classify Twitter bots dataset as harmful and non-harmful bots and further compare their results using evaluation metrics. Equations 4, 5, 6, and 7, respectively are the mathematical representation of the machine learning algorithms employed in this research.

3.2.1 Catboost Classifier

A gradient-boosting technique called CatBoost uses binary decision trees as its basis predictor. Consider some observed data and samples $G = \{(Q_i, P_j)\}\, i = 1_{,\ldots,m,}$ where $X_j = (q_1^1, q_1^2, q_1^n)$ is a vector that includes the response feature, and n features $P_z \in S$, which can be encoded as a number feature or binary (i.e., yes or no) (0 or 1). Samples $(P_z \in S)$are uniformly dispersed based on an unidentified distribution $(.,.)$. The learning task's objective is to develop a function $H: S^n \to S$ that minimizes the expected loss specified as

$$J(H): EJ(p, H(Q)) \quad (4)$$

where (Q, p) is a sample of the test data taken from the training data D and $J(.,.)$ is a smooth loss function. The gradient boosting process creates a series of approximations iteratively $H^t: S^m \to S, t = 0, 1,$ in a greedy fashion. From the previous approximation $H^{t-1}, H^t$is obtained in an additive process, such that $H^t = H^{t-1} H\alpha g^t$, with a step size α and function g^t: $R^n \to R$, which is a base predictor, is selected from a set of functions G in order to reduce or minimize the expected loss as given thus:

$$g^t = \text{argmin}_{g \in G} J(H^{t-1} + g) = \text{argmin}_{g \in G} EJ(y, H^{t-1}(X) + g(X)) \quad (5)$$

The Newton technique frequently employs a second-order approximation to the minimization issue $J(H^{t-1} + g^t)$J(Ht−1 +g t) atH^{t-1} or by performing a step of the (negative) gradient. A gradient descent function is one of these.

3.2.2 Logistic Regression(LR)

With the use of a Logistic Function, Logistic Regression happens to be a statistical model which is frequently used to model a binary dependent variable. It can be utilized for multiclass classification as well as tasks that need binary classification[16]. Logistic functions are sometimes known as a sigmoid function, and it is technically denoted by:

$$F(x) = \frac{1}{1+e^{-x}} = \frac{e}{e^x+1} \quad (6)$$

3.2.3 Random Forest Algorithm (RF)

A Random Forest is an ensemble of Classification and Regression Trees created by randomly resampling the training set and training on datasets of the same size. A collection of bootstraps is used as a test set after the tree has been built, excluding any particular record from the initial dataset [out-of-bag (OOB) samples]. The classification error rate across all test sets serves as the OOB estimate of the generalization error. The OOB error is precise when using a test set that is the same size as the training set. It is unnecessary to use a separate test set if the OOB estimate is used instead. Each tree selects one of the classes to be categorized, and the forest predicts the class with the highest votes.

The concept of variable importance, in which Random Forest selects simplicity using a random subspace methodology, is analyzed using the Gini impurity criteria index [17][18]. The Gini index is a gauge of a variable's ability to predict outcomes in a regression or classification model and is based on the concept of impurity reduction. As a result of its non-parametric nature, it is not dependent on data

from a particular type of distribution. Assuming a binary split, the Gini index of node n is calculated as follows:

$$\text{Gini}(c)\sum_{k=1}^{2}(q_i)^2 \quad (7)$$

where q_i is class j relative frequency in the node n.

The $Giniindex$ should be improved to the greatest extent possible while splitting a binary node. In other words, a low $Gini$ (which means a bigger fall in Gini) shows that a specific predictor trait is more important in dividing the data into two classes. The $Giniindex$can be used to rank the importance of features in a classification process.

3.2.4. Evaluation metrics

Evaluation metrics are the standard metrics used in checking the efficacy of an algorithm to know its performance on a given dataset [19]. Equations 8, 9, 10, and 11 are the standard evaluation metrics used in comparing the results of the machine learning models employed in this research.

$$Accuracy = \frac{TP}{TP + TN + FP + FN} (8)$$

$$Precison = \frac{TP}{TP + FP} (9)$$

$$Recall = \frac{TP}{(TP) + (FN)} (10)$$

$$F1\ Score = 2 * \frac{Precison * Recall}{Precison + Recall} (11)$$

where TP is the True Positive, FP is the False Positive, TN is the True Negative, and FN is the False Negative.

4. Results and Discussion

4.1 Results

The initial dataset consisting of 11960 instances, with 8 features, was pre-processed to get the desired outcome for the experiments. During the preprocessing stage, it was discovered that many of the features were not relevant to the research; the features were then reduced from 8 to 2 features (tweet and type) using a decision tree. In selecting the relevant features, about 181 instances were not among the records with the selected features, which were further deleted from the dataset to be left with those having the relevant features. Therefore, out of the 11968 instances that were present, it was reduced to 11787 instances, distributed into 5983 and 5804 normal and attack instances, respectively, where 80% of the dataset was used for training and 20% for testing, as presented in Table 3.

Table 3: Pre-process data distribution

Instances		**CLASS**	
		Not harmful (0)	**Harmful (1)**
Train (80%)	9429	4764	4665
Test (20%)	2358	1219	1139
Total	11787	5983	5804

The two categories are labeled as 0, and 1 for normal and attack classes, respectively. After eliminating the irrelevant records, thereafter, the algorithms were trained and evaluated to check their efficacy on both the training and testingThe matrix used in evaluating the performance of the models on both training and testing are True Positive (TP) indicating correctly predicted harmful bots, False Positive (FP) indicating non harmful bots misclassified as harmful, True Negative(TN) indicating correctly classified non-harmful bots, and False Negative (FN) which represents harmful bots misclassified as non-harmful,F1 Score, Accuracy, Precision, Recall, and F1_score. The implementation was done using python programming language. The confusion matrix of the models from python library on testing and training sets are displaced in Figure 2, 3 and 4 respectively, while Table 4 and 5 are the overall performance of the models with their corresponding graphs displayed in Figure 5 and 6, respectively.

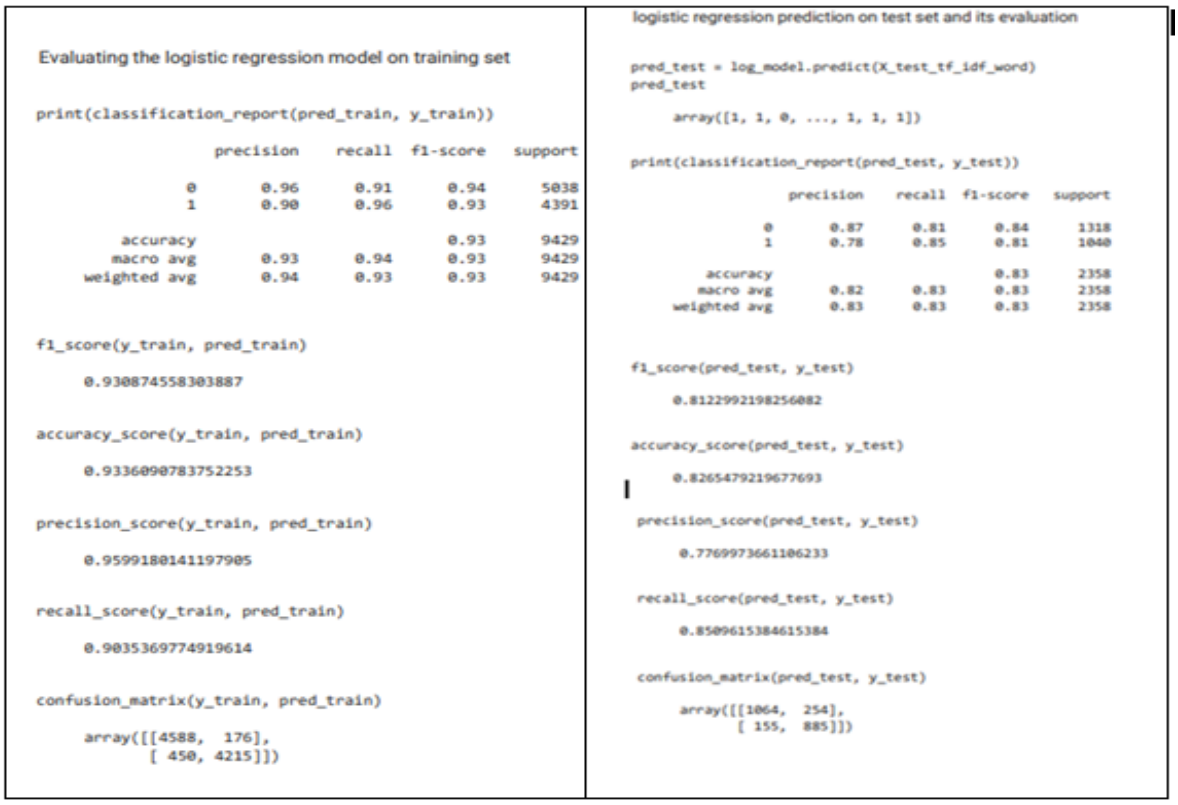

Figure 2: Logistic Regression Confusion Matrix

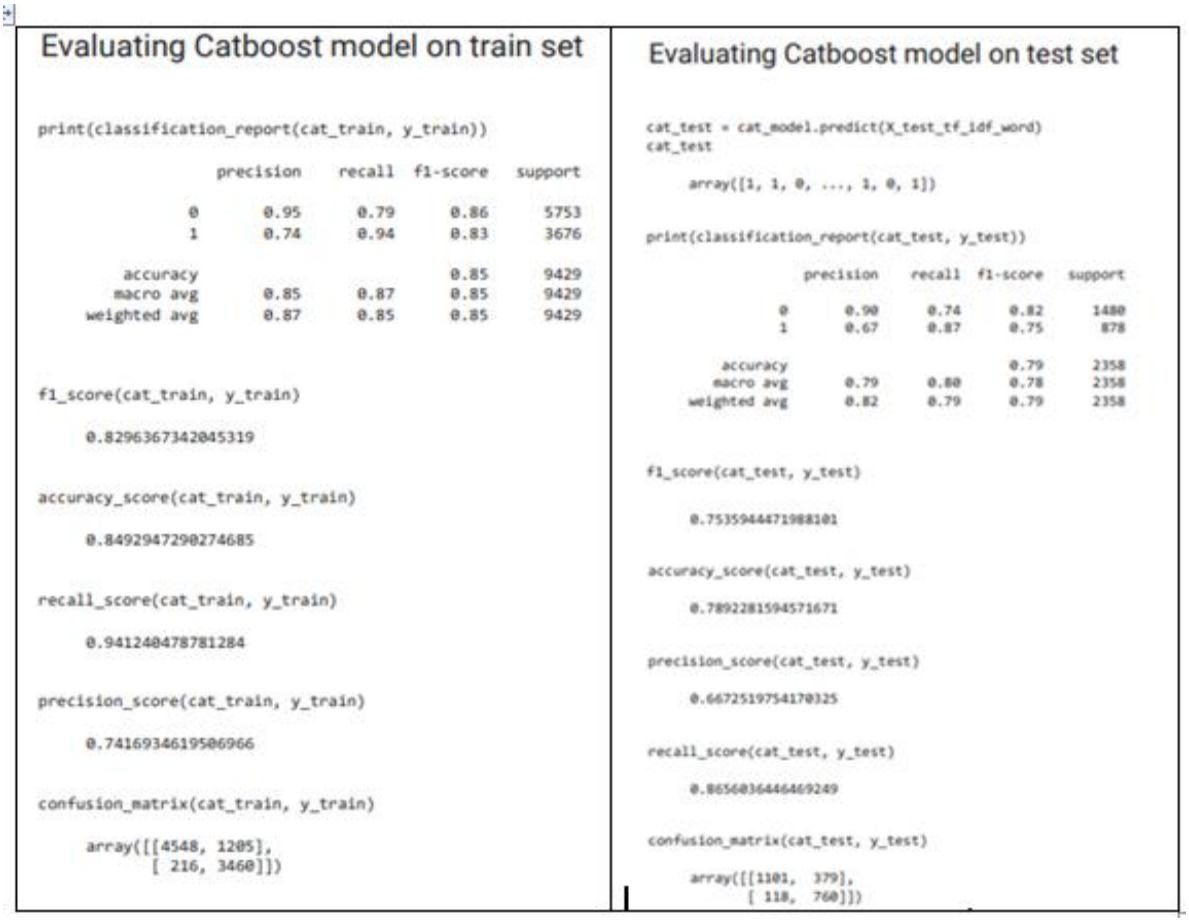

Figure 3: Catboost Confusion Matrix

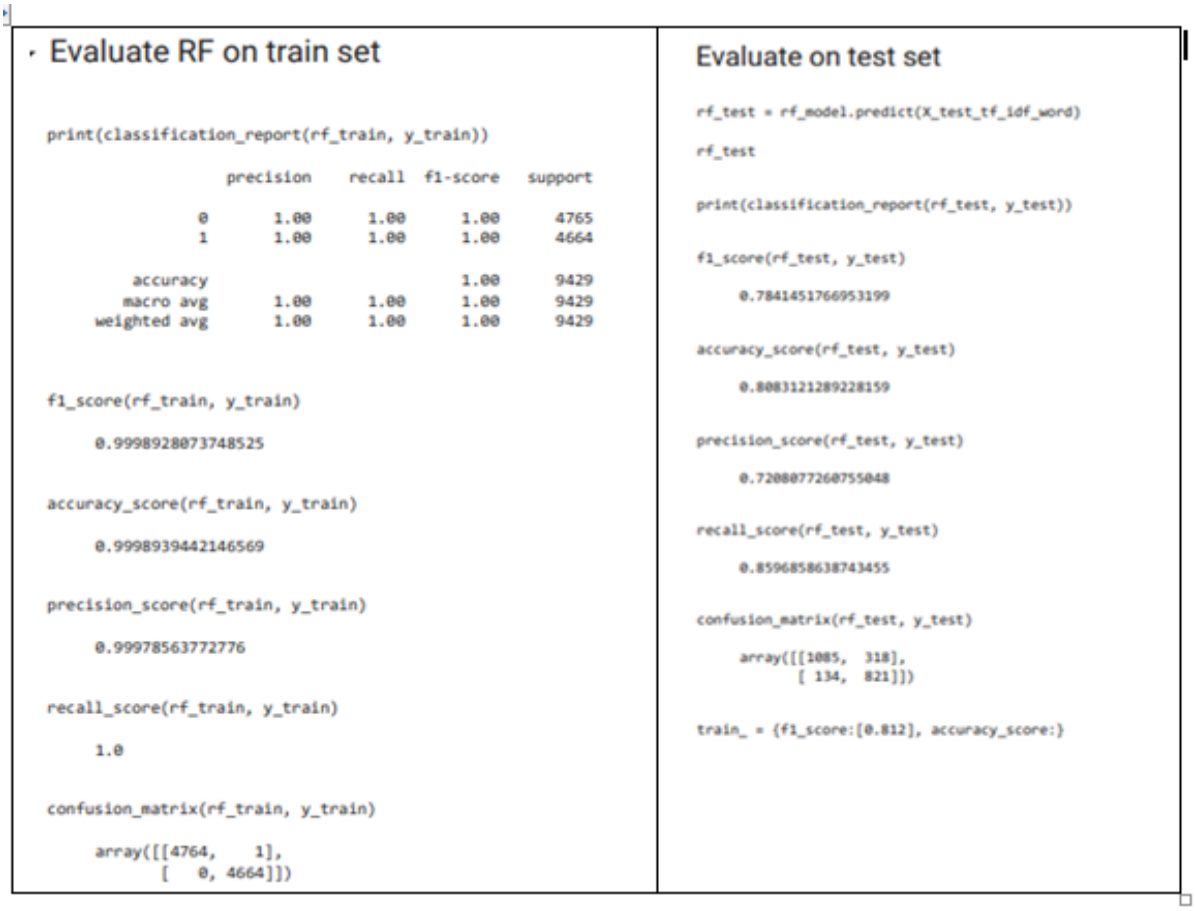

Figure 4: Random Forest Confusion Matrix

Table 4: Performance evaluation of the models on training set

Classifiers	**TP**	**FP**	**FN**	**TN**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	4588	176	450	4215	93%	96%	90%	93%
Catboost	4548	1205	216	3460	85%	74%	94%	83%
Random Forest	4764	1	0	4664	99%	99%	100%	99%

Table 5: Performance evaluation of the models on test set

Classifiers	**TP**	**FP**	**FN**	**TN**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	1064	254	155	885	83%	78%	85%	81%
CatBoost	1101	379	118	760	79%	67%	87%	75%
Random Forest	1085	318	134	821	81%	72%	86%	78%

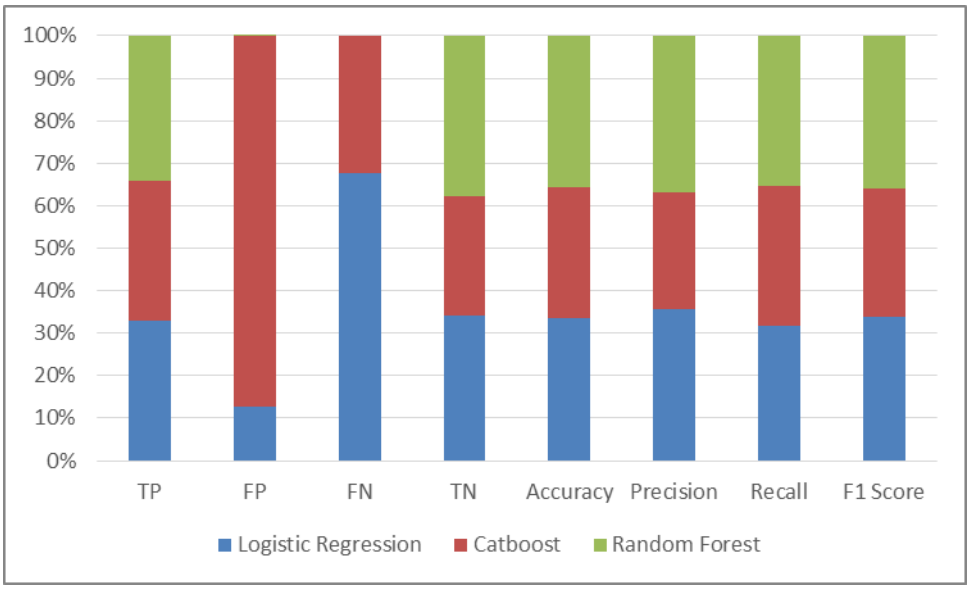

Figure 5: Graphical representation of the models on train set

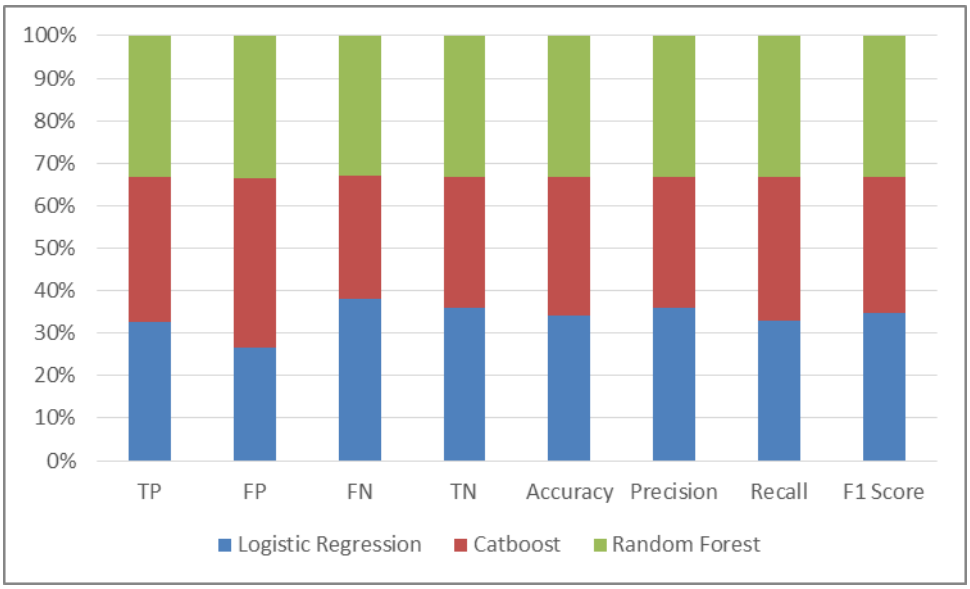

Figure 6: Graphical representation of the models on test set

4.2 Discussion

The LR model displayed in Figure 2 recorded accuracy of 93% and 83% on both training and testing sets respectively. For the confusion matrix, out of 4764 not harmful bots instances that were present in the training set, 4588 instances were correctly classified as not harmful while 176 were misclassified as harmful. The number of correctly classified harmful bots instances were 4215 while 450 instances were misclassified as not harmful out of 4665. On the test set, 1064 instances were correctly classified as not harmful while 254 were misclassified as harmful out of 1219 of not harmful bots instances. The total number of harmful bots present in the test set were1139 instances out of which 885 were correctly classified and 155 were misclassified. In summary, the model recorded approximately93%, 96%, and 90%, and 81%, 78%, and 85% of F1_score, precision, and Recall on both sets.

For the Catboost presented in Figure 3, the confusion matrix recorded 4548 as correctly classified not harmful bots, and 216 as wrongly classified harmful bots, while 3460 were correctly classified as harmful and 1205 were misclassified as not harmful on the training set. On the other hand, for the test set, 1101 instances were correctly classified as not harmful, 118 were misclassified as harmful, while 760 were correctly classified as harmful and 379 were misclassified as not harmful. The model recorded approximately 85%, 74%, 94%, and 83%, and 79%, 67%, 87%, and 75%Accuracy, precision, Recall, and F1_score on both sets, respectively.

The confusion matrix for RF correctly classified all the 4764 of the not harmful, while only 1 was misclassified as not harmful out of the 4665 harmful bots that were present in the training set. For the testing set,out of 1219 not harmful bots, 1085 were correctly classified, 138 were misclassified as harmful, while 821 were classified correctly as harmful, and 314 as not harmful out of 1139 harmful bots instances in the test set. The RF model recorded approximately 81%, 72%, 86%, and 78%, and 99%, 99%, and 100%, and 99 of Accuracy, precision, Recall, and F1_score on both sets, respectively.

5. Conclusion

We have successfully applied various machine learning techniques to classify Twitter bots as harmful and non-harmful. The efficacy of the models were tested on both training and testing sets of Twitter dataset. In comparing the models, RF classifier recorded the highest performance on training, while LR achieved highest performance in terms of accuracy, precision, and F1 Score, respectively. Our findings have shown that the classification systems can help in identifying and mitigate the impact of harmful bots on Twitter . In this study, the dataset gotten has few features that determine the target variable. In future works, it is recommended to have a large Twitter bots dataset combined with the application of different feature extraction and classification methods to provide promising and robust features for the model to learn from.

References

[1] Jalal, N., & Ghafoor, K. Z. (2022). Machine learning algorithms for detecting and analyzing social bots using a novel dataset. ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY, 10(2), 11-21.

[2] Chawla, V., & Kapoor, Y. (2023). A hybrid framework for bot detection on twitter: Fusing digital DNA with BERT. Multimedia Tools and Applications, *82*(20), 30831-30854.

[3] Ramalingaiah, A., Hussaini, S., &Chaudhari, S. (2021, August). Twitter bot detection using supervised machine learning. In Journal of Physics: Conference Series (Vol. 1950, No. 1, p.012006). IOP Publishing.

[4] Alarfaj, F. K., Ahmad, H., Khan, H. U., Alomair, A. M., Almusallam, N., &Ahmed, M. (2023). Twitter bot detection using diverse content features and applying machine learning algorithms. *Sustainability*, *15*(8), 6662.

[5] Aqilah Aini Zahra, Widyawan, and SilmiFauziati "Bot Detection Applicatoon on Twitter Using Machine Learning with Random Forrest Classifier Algorithm" Vol. 4, No. 2, pp 1-2, June 2020

[6] Hayawi, K., Mathew, S., Venugopal, N., Masud, M. M., & Ho, P. H. (2022). DeeProBot: a hybrid deep neural network model for social bot detection based on user profile data. Social Network Analysis and Mining, 12(1), 43.

[7] Heidari, M., James Jr, H., & Uzuner, O. (2021, April). An empirical study of machine learning algorithms for social media bot detection. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-5). IEEE.

[8] Ilias, L., & Roussaki, I. (2021). Detecting malicious activity in Twitter using deep learning techniques. Applied Soft Computing, 107, 107360.

[9] Martín-Gutiérrez, D., Hernández-Peñaloza, G., Hernández, A. B., Lozano-Diez, A., &Álvarez, F. (2021). A deep learning approach for robust detection of bots in twitter using transformers. IEEE Access, 9, 54591-54601.

[10] 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.

[11] Liu, X., Zhan, Y., Jin, H., Wang, Y., & Zhang, Y. (2023). Research on the classification methods of social bots. Electronics, 12(14), 3030.

[12] Khaled, S., El-Tazi, N., & Mokhtar, H. M. (2018, December). Detecting fake accounts on social media. *In 2018 IEEE internationalconference on big data (big data)* (pp. 3672-3681). IEEE.

[13] Wei, F., & Nguyen, U. T. (2019, December). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In 2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA) (pp. 101-109). IEEE.

[14] Nguyen, M., Aktas, M., & Akbas, E. (2020). Bot detection on social networks using persistent homology.Mathematical and Computational Applications, 25(3), 58.

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

[16] Kushta, E., & Trushaj, G. (2020). Implementation of The Logistic Regression Model and Its Applications. Journal of Advances in Mathematics, *18*, 46-51.

[17] Strobl, C., Boulesteix, A. L., Zeileis, A., &Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, *8*(1), 1-21.

[18] Ceriani, L., & Verme, P. (2012). The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. The Journal of Economic Inequality, 10(3), 421- 443.

[19] Ayogu, B. A., Ogunleye, G. O., Benjamin, T. F., & Ugwu, J. N. (2023). Performance Evaluation of Hybrid and Standalone Techniques on Web Applications Based Cross-Site Scripting Attacks. *LAUTECH Journal of Engineering and Technology*, *17*(2), 111-120.