International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 4
ISSN: 2088-5334

X Bot Detection Using One-Class Classification Methods with Isolation Forest Algorithm

Yusup Miftahuddin [a,*], Muhammad Haydar Al-Ghifary [a]

[a] *Department of Informatics, Bandung National Institute of Technology (Itenas), Jl. Phh. Mustofa No. 23, Bandung, Indonesia*
*Corresponding author: *yusufm@itenas.ac.id*

***Abstract*—X bots pose a significant issue in the social media landscape, with many shared links originating from bot-like accounts. This study introduces the application of the Isolation Forest algorithm, aimed explicitly at identifying anomalies such as bots by analyzing X account details. This study utilizes a dataset that merges data from Botometer with supplementary metrics like 'average tweets per day' and 'account age in days', contributed by David Martín Gutiérrez. This approach was adopted due to the increasing difficulties accessing the X API. The dataset comprises 37,438 instances, with 25,013 labeled human accounts and 12,425 labeled bot accounts. Pre-processing is performed to remove irrelevant features, and the dataset is split into Training, Validation, and Test sets in a 70:15:15 ratio. The training set undergoes hyperparameter and threshold tuning to identify the best configuration for this specific dataset (n_estimators: 50, contamination: 0.5, bootstrap: True), achieving a training set F1-score of 0.211001. Despite these optimization efforts, the Isolation Forest model's performance remains relatively low. The Test set evaluation yields modest precision, recall, and F1-score values (0.1801, 0.2795, and 0.2190, respectively), with a ROC AUC score of 0.3272. While the Isolation Forest algorithm shows promise in detecting X bots, its performance on this specific dataset is limited. Isolation Forest may not be the most suitable algorithm for this particular bot detection task on this dataset. Future work will explore techniques to enhance the performance of bot detection for a more comprehensive analysis.**

***Keywords*— X; bot detection; anomaly; one class classification; isolation forest.**

Manuscript received 8 Aug. 2023; revised 22 Apr. 2024; accepted 20 May 2024. Date of publication 31 Aug. 2024.

I. INTRODUCTION

Artificial Intelligence (AI) remains a prominent trend in the rapidly advancing world of technology. AI is used to mimic human behavior and thinking patterns, one of which is by implementing bots. Bots are automated systems that perform tasks repetitively and efficiently, making them more effective than their human counterparts in consistent and tireless execution [1]-[5]. As a social media platform, X allows bots to be used, known as Xbots. These automated X accounts serve various purposes, such as sending automatic tweets, following other accounts, or responding to tweets automatically. While Xbots have positive applications, they can also be misused for damaging purposes, including spamming and spreading hoaxes [6], [7].

Xbots have become a significant problem in the social media ecosystem. A study conducted by the Pew Research Center in 2018 utilized a tool called *Botometer* to estimate the proportion of X links leading to popular websites posted by automated or partially automated accounts [8]. The study revealed that approximately 66% of all shared X links originated from accounts exhibiting characteristics commonly associated with bots or automated accounts rather than human users. Additionally, Research conducted by Chu et al. [1] can identify human, cyborg, or bot accounts by observing differences in tweeting behavior habits, the content of the tweets, and account characteristics such as the number of followers, following, and retweets. On the content of tweets, emotion/emotional sentiment can be detected using machine learning or polarity [1], [9]-[15].

The high prevalence of bots negatively impacts the integrity of information and user experience on the platform. Considering the upcoming elections during this research, the importance of accuracy, transparency, and protection against manipulation and disinformation is crucial. Therefore, as conducted in this thesis, this research on bot detection on X will contribute to building X as a more transparent and trustworthy social media platform while preserving its integrity during critical events such as elections.

To address this challenge, this study employs the One-Class Classification (OCC) method, specifically the Isolation Forest algorithm, to detect bots on X. The Isolation Forest algorithm, proposed by Liu et al. [16], is an approach used to identify and isolate anomalies or rare data points from standard data [16]-[25]. This research aims to efficiently classify X accounts as bots or non-bots based on their behavior patterns by applying OCC with the Isolation Forest algorithm. This approach allows for accurate and automated identification of suspicious accounts with abnormal behavior, contributing to building a more transparent and trustworthy X platform during elections or other significant events.

II. Materials and Methods

Fig. 1 illustrates the system-building process, starting with Dataset input and followed by pre-processing. In the pre-processing step, the data will be cleaned and feature selection by removing non-numeric data because Isolation Forest can only count the numerical. The data will go through the data normalization using L2-Norm. After that, the data will be split by 70:30 for training and testing. The training process will go through the Isolation Forest model making, including making the Isolation Trees, Anomaly Score Calculation, and Identifying the anomaly. After that, the same model will be used in the test process.

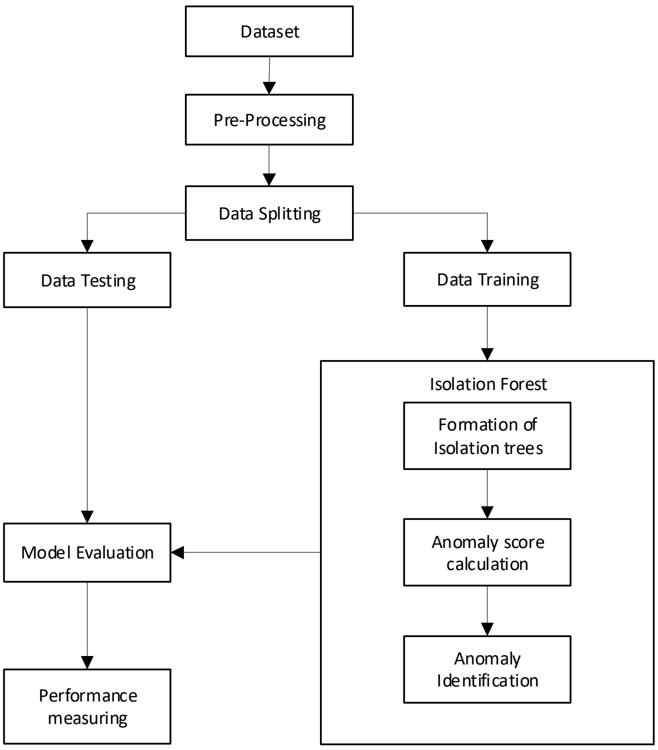

Fig. 1 Block Diagrams of the system

A. Data Preparation

This study utilizes a Botometer and X Bot Repository dataset, primarily curated by David et al. [26]. As X's free API access became increasingly restrictive, additional data was collected to enrich the dataset. The dataset, comprising 37,438 entries, each represents a unique X account. Each entry includes the X ID and a target variable, 'account_type,' indicating whether the account is a 'bot' or 'human.' Of these accounts, 25,013 are labeled as human and 12,425 as bots. Botometer, a machine learning tool, was used to assign bot scores between 0 and 1 to each account based on an analysis of approximately 1200 account-related features. Accounts with higher scores, indicative of bot-like activity, were labeled as bots, while those with lower scores were labeled as human.

The dataset amalgamates several smaller datasets from previous investigations into suspicious X accounts. Using the identifiers from these datasets, account data was retrieved via the X API. The resulting dataset is a more streamlined and comprehensive version of its predecessors, designed to enhance analysis. Inactive X accounts were excluded from the dataset, and the information for the remaining accounts was updated based on data available as of July 13, 2020.

B. Pre-Processing and Feature Selection

Feature selection in this study is based on relevant attributes for detecting bot accounts on X, as identified in previous research by Davis et al. [27], and Varol et al. [28]. The following steps are applied for feature engineering:

1) Dropping Unwanted Columns: Columns such *'Unnamed: 0', 'created_at', 'description', 'lang', 'location', 'profile_background_image_url', 'profile_image_url',* and *'screen*_name' are dropped from the dataset as they are not directly relevant to predicting bot accounts.

2) The remaining columns are rearranged in a meaningful order, including features like *'id', 'default_profile', 'default_profile_image', 'favourites_count', 'followers_count', 'friends_count', 'geo_enabled', 'statuses_count', 'verified', 'average_tweets_per_day', 'account_age_days',* and *'account_type'.* This reordering facilitates better data organization and prioritizes essential features for analysis.

3) Converting Boolean Values: Columns with Boolean values ('True' or 'False'), such as *'verified'*, *'default*_profile', 'default_profile_*image'*, and 'geo_*enabled'*, are converted to numeric representation ('True' → 1, 'False' → 0) to ensure compatibility with the Isolation Forest model.

The comprehensive details of the pre-processing and feature selection steps are encapsulated in Table I.

C. Data Splitting

The dataset was initially labeled as supervised to ensure a robust evaluation of the Isolation Forest model's performance. It was transformed into an unsupervised setting by splitting it into three subsets: the training set, validation set, and holdout test set. The training set was utilized to train the model using various hyperparameter configurations, while the validation set was employed to fine-tune the thresholds for each configuration. Subsequently, the model's effectiveness in detecting bot accounts was evaluated on the holdout test set comprising unseen data to validate its performance. The dataset was partitioned into a 70:15:15 ratio to facilitate the training, validation, and testing. The Training Set is shown in Table II, The Validation Set is shown in Table III, and the Test Set is shown in Table IV.

TABLE I
DATASET AFTER PRE-PROCESSING AND FEATURE SELECTION

No	Id	Default Profile	Default Profile Image	Favorites Count	Followers Count	Friends Count	Geo Enab led	Status Count	Verified	Average Tweets per Day	Accoun t Age Days	Account type
0	8E+17	0	0	4	1589	4	0	11041	0	7.870	1403	1
1	8E+17	0	0	536	860	880	0	252	0	0.183	1379	0
2	9E+17	0	0	3307	172	594	1	1001	0	0.864	1159	0
3	8E+17	1	0	8433	517	633	1	1324	0	0.889	1489	0
4	5E+08	0	0	88	753678	116	1	4202	1	1.339	3138	0
...	...	...	...	...	...	...	...	...	...	...	...	...
37433	6E+07	1	0	651	139	1105	0	340	0	0.084	4028	0
37434	1E+09	0	0	8839	1121486	605	1	24970	1	8.976	2782	0
37435	1E+09	1	0	399	85630	190	0	6174	1	2.226	2773	0
37436	8E+08	0	0	967	138	166	1	982	0	0.339	2899	0
37437	4E+08	0	0	1092	5	39	0	1563	0	0.493	3172	1

TABLE II
DATA SPLITTING – TRAINING SET

No	Id	Default Profile	Default Profile Image	Favorites Count	Followers Count	Friends Count	Geo Enabled	Status Count	Verified	Average Tweets Per Day	Account Age Days
6462	1E+07	0	0	82605	474780	90669	1	92773	1	20.607	4502
33743	2E+09	1	1	1731	9	0	0	1730	0	0.751	2304
3668	4E+09	1	0	7986	562	2076	0	1901	0	1.092	1741
24145	1E+07	0	0	2152	20434	5009	0	25785	1	5.765	4473
22772	9E+08	0	0	0	43	410	0	1648	0	0.578	2852
...	...	...	...	...	...	...	...	...	...	...	...
16850	2E+09	1	0	1753	144	167	0	3028	0	1.252	2419
6265	2E+08	1	0	396	4	0	0	583	0	0.160	3640
11284	8E+17	1	0	1301	4	80	0	1938	0	1.431	1354
860	6E+08	1	0	17796	405	453	1	32900	0	10.876	3025
15795	3E+07	0	0	11190	381932	3648	0	100691	1	24.269	4149

TABLE III
DATA SPLITTING – VALIDATION SET

No	Id	Default Profile	Default Profile Image	Favorites Count	Followers Count	Friends Count	Geo Enabled	Status Count	Verified	Average Tweets Per Day	Account Age Days
1072	4E+08	0	0	6315	248	125	0	4901	0	1.548	3167
26845	1E+08	0	0	504	13798	829	1	2106	0	0.561	3756
2726	4E+09	0	0	22590	490	380	1	39463	0	22.059	1789
15091	2E+07	1	0	121	558956	665	0	10186	1	2.439	4176
34296	8E+17	0	0	134	127	248	0	13318	0	10.421	1278
...	...	...	...	...	...	...	...	...	...	...	...
31630	8E+17	0	1	1822	74	76	0	959	0	0.645	1487
37027	4E+09	1	0	16735	12127	954	1	34314	1	19.343	1774
31204	2E+09	0	0	60	21	58	0	1102	0	0.449	2453
27298	2E+09	1	0	8419	432	648	1	2948	0	1.202	2452
8274	4E+08	0	0	3742	193	273	0	1397	0	0.441	3165

TABLE IV
DATA SPLITTING – HOLDOUT/TEST SET

No	Id	Default Profile	Default Profile Image	Favorites Count	Followers Count	Friends Count	Geo Enabled	Status Count	Verif ied	Average Tweets Per Day	Account Age Days
6053	3E+09	1	0	762	8	90	0	1600	0	0.718	2227
35865	5E+08	1	0	1695	3	0	0	2325	0	0.741	3137
4104	4E+08	1	0	16	23	0	1	407	0	0.124	3282
13729	4E+08	1	0	247	17	0	0	44	0	0.014	3190
2924	3E+09	1	0	17424	293	135	1	23104	0	11.680	1978
...	...	...	...	...	...	...	...	...	...	...	...
35447	5E+07	0	0	305502	23492	25181	1	569784	0	140.341	4060
36756	4E+08	0	0	843	43220	4278	1	6487	1	1.997	3248
15192	1E+09	0	0	1836	44678	1211	1	10993	1	3.999	2749
9081	3E+08	0	0	1946	17904	117	1	3651	1	1.052	3472
25067	9E+17	1	0	141	0	0	0	188	0	0.173	1088

D. Isolation Forest

The Isolation Forest algorithm is an unsupervised learning technique used for anomaly detection. Anomaly detection is a critical task in various domains, aiming to identify rare, unusual, or abnormal patterns in data that deviate significantly from the majority or normal behavior. Anomalies are often indicative of potential issues, fraudulent activities, or critical events in the data [29]. One-Class Classification (OCC) and the Isolation Forest algorithm are two powerful techniques used for detecting anomalies in unsupervised settings where labeled anomaly data is scarce or unavailable [21].

Anomaly detection involves identifying data instances that exhibit exceptional behavior compared to the majority of the data. These anomalies are data points that do not conform to the expected patterns or distribution of the normal data. Anomalies can represent valuable insights or critical events, such as fraudulent transactions, system faults, or emerging threats.

Isolation Forest efficiently isolates anomalies by randomly partitioning data points into binary trees. The algorithm measures the average path length required to isolate a data point, allowing it to identify anomalies as points that can be isolated in fewer splits compared to standard data points [30].

- Path Length: The path length ($h(x)$) of a data point x in the tree is defined as the number of edges traversed from the root node to reach the terminal node (anomaly score).
- Average Path Length: The average path length (c(n)) for a tree with n data points is calculated as

$$c(n) = 2 \times (\log n - 1) - \frac{2\times(n-1)}{n} \quad (1)$$

- Anomaly Score: The anomaly score (s(x)) for a data point x is determined as

$$s(x) = 2^{-\frac{h(x)}{c(n)}} \quad (2)$$

- Threshold: A threshold distinguishes between normal and abnormal data points. Data points with an anomaly score below the threshold are classified as anomalies

The parameters used in the Isolation Forest model are shown in Table V.

1) Hyperparameter and Threshold Tuning: For this research, the Python library Scikit-Learn from the *sklearn* package will be utilized. It offers a comprehensive set of parameters listed in the following table.

TABLE V
ISOLATION FOREST PARAMETERS

Parameter	Description	Range or Values
bootstrap	True: Trees fit on random subsets with replacement; false: Sampling without replacement.	Boolean (default = False)
contamination	The proportion of outliers in the data set defines the threshold on sample scores.	Float, auto (default)
max_features	Number of features to draw from X to train each base estimator.	int, float (default=1.0)
max_samples	All samples used if larger than provided samples (no sampling).	auto (default), int or float
n_estimator	Number of base estimators in the ensemble.	int, 100 (default)
n_jobs	Number of jobs to run in parallel for fit and predict.	int (default=None)
random_state	Controls pseudo-randomness of feature and split value selection.	int, RandomState instance (default=None)
verbose	Controls the verbosity of the tree building process.	int (default=0)
warm_start	True: Reuse solution of previous fit and add more estimators; False: Fit a whole new forest.	Boolean (default=False)

This research aims to optimize the Isolation Forest algorithm for precise bot account detection on X. The focus is on two key aspects: hyperparameter tuning and threshold selection. By fine-tuning *bootstrap*, *contamination*, and *n_estimators*, the goal is to achieve maximum anomaly detection precision. Different threshold values are also explored to balance precision and recall, aiming to maximize the F1 score for identifying bot accounts.

The F1 score is a performance metric to evaluate the model's accuracy in detecting positive and negative instances. It considers both precision and recall to provide a balanced measure of the model's effectiveness in anomaly detection [31]. The F1-score is calculated as follows:

$$F1 - score = \frac{2\times(Precision \times Recall)}{(Precision \times Recal)} \quad (3)$$

where:

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive+False Positive})} \quad (4)$$

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)} \quad (5)$$

The F1-score ranges from 0 to 1, where a higher value indicates a better balance between precision and recall, resulting in more accurate anomaly detection. Table VI shows that the Isolation Forest model has been optimized through 3030 iterations to deliver accurate and reliable results for robust anomaly detection within the X dataset.

TABLE VI
THE RESULT AFTER HYPERPARAMETER AND THRESHOLD TUNING

Index	n_estimators	contamination	bootstrap	Threshold	F1-score
0	50	0.5	TRUE	0.00	0.211001
1	50	0.5	TRUE	0.01	0.211001
2	50	0.5	TRUE	0.02	0.211001
3	50	0.5	TRUE	0.03	0.211001
4	50	0.5	TRUE	0.04	0.211001
...	...	...	...	...	...
3025	100	0.1	FALSE	0.96	0.114868
3026	100	0.1	FALSE	0.97	0.114868
3027	100	0.1	FALSE	0.98	0.114868
3028	100	0.1	FALSE	0.99	0.114868
3029	100	0.1	FALSE	1.00	0.114868

Table VI presents exhaustive results of hyperparameter tuning, showcasing F1 scores for each parameter and threshold combination during cross-validation. This provides valuable insights into model performance under different settings. Table 8 displays the 30 unique F1 scores observed across the iterations, illustrating variations in precision based on various parameter combinations. Notably, the highest recorded F1 score of 0.211001 at iteration 8 represents the optimal performance achievable by the Isolation Forest model on this dataset.

Furthermore, the data has been visualized in Figure 2 and Figure 3 to provide a clear overview. The charts indicate that the combination of 50 Estimators and Contamination 0.5 yields the highest F1-score of 0.21. Additionally, the bootstrap parameter has been set to True to achieve this optimal performance in detecting bot accounts on X.

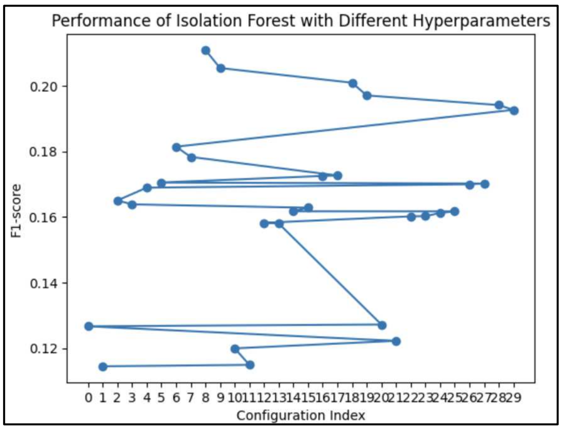

Fig. 2 Line Chart of F1-Score

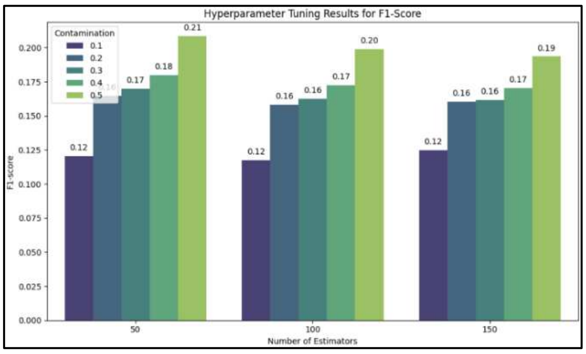

Fig. 3 Charts for F1-Score based on the Parameters

TABLE VII
THE UNIQUE VALUES IN HYPERPARAMETER AND THRESHOLD TUNING

No	n_estimators	contamination	bootstrap	Threshold	Precision	Recall	F1-score
8	50	0.5	TRUE	0.01	0.174	0.262	0.211
9	50	0.5	FALSE	0.01	0.172	0.259	0.205
18	100	0.5	TRUE	0.01	0.163	0.245	0.200
19	100	0.5	FALSE	0.01	0.164	0.246	0.197
28	150	0.5	TRUE	0.01	0.162	0.243	0.194
29	150	0.5	FALSE	0.01	0.162	0.244	0.193
6	50	0.4	TRUE	0.01	0.155	0.186	0.181
7	50	0.4	FALSE	0.01	0.160	0.193	0.178
17	100	0.4	FALSE	0.01	0.153	0.184	0.173
16	100	0.4	TRUE	0.01	0.151	0.182	0.173

2) *Testing the Model Performance*: Following the best hyperparameters and optimal threshold determination, the Isolation Forest model is implemented on the test set for evaluation. The model is first trained using the training set and then utilized to predict the test set. The resulting predictions undergo additional validation, where data points with scores equal to 1 are designated as Inliers (non-bots), while scores of -1 correspond to Outliers (bots). The evaluation outcomes are depicted in Table VIII.

TABLE VIII
THE RESULT AFTER IMPLEMENTING MODEL TO TEST DATA

No	id	account type	anomaly scores	result
0	2,6E+09	Bot	0.029269	Non-Bot
1	4,7E+08	Bot	0.035162	Non-Bot
2	3,6E+08	Non-Bot	0.008399	Non-Bot
3	4,2E+08	Bot	0.035264	Non-Bot
4	3,1E+09	Non-Bot	-0.031691	Bot
...	...	...	...	...
5611	5,5E+07	Non-Bot	-0.329722	Bot
5612	3,8E+08	Non-Bot	-0.039911	Bot
5613	1,2E+09	Non-Bot	-0.034016	Bot
5614	2,5E+08	Non-Bot	-0.023639	Bot
5615	9E+17	Non-Bot	0.017579	Non-Bot

In this phase, the Isolation Forest model on the test set is evaluated, and the following metrics are as obtained:

- Precision: 0.18007662835249041
- Recall: 0.27945945945945944
- F1-score: 0.2190213937725058
- ROC AUC Score: 0.32719653801438187

The count of correctly predicted 'Bot' accounts (True Positives) is 517, and the count of correctly predicted 'Non-Bot' accounts (True Negatives) is 1412. These evaluation metrics and counts provide valuable insights into the model's ability to detect bot accounts within the X dataset accurately. Despite the optimization efforts, the Isolation Forest model's performance on the test set yields relatively low scores, indicating that it may not be the most suitable algorithm for this specific task.

III. RESULTS AND DISCUSSION

A. *Isolation Forest Results*

The Isolation Forest results analyze the model’s performance using various graphical representations. The Confusion Matrix, as shown in Figure 4, provides a detailed breakdown of the model’s predictions.

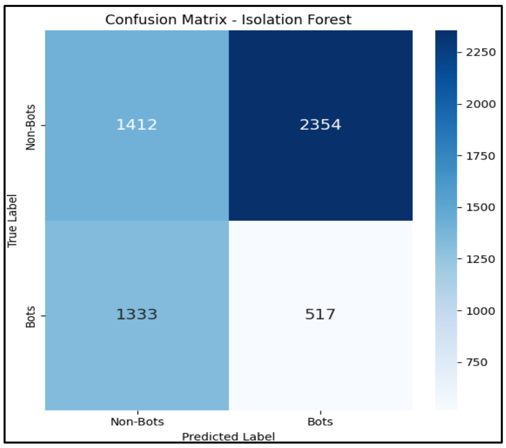

Fig. 4 Confusion Matrix of the Isolation Forest Process

- True Non-Bots (TN): There are 1412 instances of non-bot accounts correctly predicted as "Non-Bots."

- False Bots (FP): There are 2354 instances of non-bot accounts incorrectly predicted as "Bots."
- False Non-Bots (FN): 1333 bot accounts are incorrectly predicted as "Non-Bots."
- True Bots (TP): There are 517 instances of bot accounts that are correctly predicted as "Bots."

Additionally, the ROC curves for both Anomaly Scores and Binary Predictions are presented in Figures 5 and 6. ROC AUC (Receiver Operating Characteristic – Area Under the Curve) is another evaluation metric used to assess the model's ability to distinguish between positive and negative instances [30], [32]–[34]. It measures the area under the receiver operating characteristic curve, which plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold values.

The ROC curve for Anomaly Scores illustrates the model's ability to distinguish between bot and non-bot accounts based on the calculated anomaly scores. On the other hand, the ROC curve for Binary Predictions shows the model's performance in classifying accounts as either "Bot" or "Non-Bot" at various threshold settings.

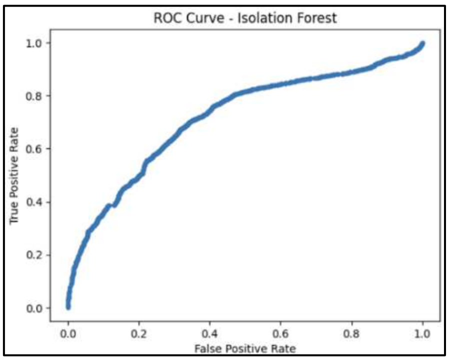

Fig. 5 ROC Curve of Anomaly Scores

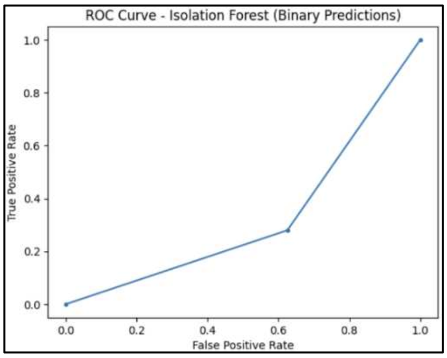

Fig. 6 ROC Curve of Binary Predictions

The ROC curves in Figure 6 and Figure 7 show that the model has a high recall but a low precision. This means the model is good at identifying all dataset bots but is also likely to locate some humans as bots. This is because the model tries to balance identifying as many bots as possible (recall) and not placing too many humans as bots (precision).

The ROC AUC score is 0.37776, which means that the model is no better than random at classifying tweets as being from bots or humans. This is because the ROC AUC score of a random model is 0.5. A higher ROC AUC score indicates that a model better distinguishes between positive and negative examples.

B. Performance Comparison

In this section, a comprehensive performance comparison is presented among five different scenarios using the same dataset: 1) Default Isolation Forest (without tuning), 2) Isolation Forest with Hyperparameter and Threshold Tuning, 3) Balanced Isolation Forest with Hyperparameter and Threshold Tuning, 4) One-Class SVM, and 5) Random Forest. The results are summarized in Table IX.

TABLE IX
PERFORMANCE COMPARISON

Model	Precision	Recall	F-1 Score	ROC AUC Score
Default Isolation Forest	0.2017	0.0768	0.1112	0.4638
Isolation Forest with Hyperparameter and Threshold Tuning	**0.1801**	**0.2795**	**0.2190**	**0.3272**
Balanced Isolation Forest with Hyperparameter and Threshold Tuning	0.1827	0.3297	0.2351	0.3025
One-Class SVM	0.3091	0.1011	0.1523	0.4950
Random Forest	0.8626	0.7632	0.8099	0.8517

Among the models evaluated, it was found that the Random Forest model outperformed the others, achieving the highest precision, recall, F1 score, and ROC AUC score. This indicates that while the Isolation Forest algorithm was the initial focus of this study, the Random Forest model demonstrated superior performance in detecting bot accounts on X. The Isolation Forest models, both default and with hyperparameter and threshold tuning, and the One-Class SVM, exhibited relatively lower precision, recall, and F1-score in this dataset. This suggests their limitations in accurately detecting bot accounts. Notably, the Isolation Forest with Hyperparameter and Threshold Tuning showed some improvement compared to the default Isolation Forest, although the improvement was not substantial.

IV. CONCLUSION

This study focused on detecting X bot accounts using the Isolation Forest algorithm, a one-class classification approach. The Isolation Forest model was optimized by fine-tuning its hyperparameters and threshold to enhance its performance in detecting bots. After an extensive evaluation, the model's best parameter configuration was identified, including *n_estimators*: 50, *contamination*: 0.5, and *bootstrap*: True.

The results indicate that the Isolation Forest model, even after hyperparameter and threshold tuning, achieved relatively low precision, recall, and F1-score, with values of 0.1801, 0.2795, and 0.2190, respectively. The ROC AUC score was also modest at 0.3272, suggesting that the model's ability to distinguish between inliers and outliers is limited.

While the hyperparameter and threshold tuning process aimed to enhance the Isolation Forest model's performance, the achieved scores remain relatively low for detecting X bot accounts. Isolation Forest may not be the most suitable algorithm for the dataset bot detection task. Further exploration of alternative models and feature engineering techniques may be necessary to achieve more accurate bot detection results.

References

[1] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting Automation of X Accounts: Are You a Human, Bot, or Cyborg?," *IEEE Trans Dependable Secure Comput*, vol. 9, no. 6, pp. 811–824, Nov. 2012, doi: 10.1109/TDSC.2012.75.

[2] D. Dukic, D. Keca, and D. Stipic, "Are You Human? Detecting Bots on X Using BERT," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, Oct. 2020, pp. 631–636. doi: 10.1109/DSAA49011.2020.00089.

[3] J. Pizarro, "Profiling Bots and Fake News Spreaders at PAN'19 and PAN'20 : Bots and Gender Profiling 2019, Profiling Fake News Spreaders on X 2020," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, Oct. 2020, pp. 626–630. doi: 10.1109/DSAA49011.2020.00088.

[4] K. Wani, A. Patil, S. Mukherjee, and S. Sarkar, "Malicious X Bot Detector," in *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, IEEE, Jan. 2021, pp. 1–6. doi:10.1109/ICNTE51185.2021.9487674.

[5] J. Arumugam, K. Lalitha, S. M. Supreetha, R. T. Shrinithi, and S. Tamilarasan, "Machine Learning For Detecting X Bot," in *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, IEEE, Jul. 2022, pp. 278–282. doi: 10.1109/CCiCT56684.2022.00059.

[6] S. Heron, "Technologies for spam detection," *Network Security*, vol. 2009, no. 1, pp. 11–15, Jan. 2009, doi: 10.1016/S1353-4858(09)70007-8.

[7] T. Bui and K. Potika, "X Bot Detection using Social Network Analysis," in *2022 Fourth International Conference on Transdisciplinary AI (TransAI)*, IEEE, Sep. 2022, pp. 87–88. doi:10.1109/TransAI54797.2022.00022.

[8] S. Wojcik, S. Messing, A. Smith, L. Rainie, and P. Hitlin, "Bots in the Twittersphere," *Pew Research Center*, Apr. 2018

[9] T. Tyagi *et al.*, "X Bot Detection using Machine Learning Models," in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2023, pp. 26–30. doi:10.1109/Confluence56041.2023.10048796.

[10] N. Narayan, "X Bot Detection using Machine Learning Algorithms," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, Sep. 2021, pp. 1–4. doi: 10.1109/ICECCT52121.2021.9616841.

[11] H. Shukla, N. Jagtap, and B. Patil, "Enhanced X bot detection using ensemble machine learning," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, IEEE, Jan. 2021, pp. 930–936. doi: 10.1109/ICICT50816.2021.9358734.

[12] F. N. Pramitha, R. B. Hadiprakoso, N. Qomariasih, and Girinoto, "X Bot Account Detection Using Supervised Machine Learning," in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE, Dec. 2021, pp. 379–383. doi:10.1109/ISRITI54043.2021.9702789.Yang

[13] S. Barhate, R. Mangla, D. Panjwani, S. Gatkal, and F. Kazi, "X bot detection and their influence in hashtag manipulation," in *2020 IEEE 17th India Council International Conference (INDICON)*, IEEE, Dec. 2020, pp. 1–7. doi: 10.1109/INDICON49873.2020.9342152.

[14] K. Sujith, S. Chowdhury, A. Goyal, A. V. Hegde, and R. Srinath, "X Bot Detection and Ranking using Supervised Machine Learning Models," in *2022 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, IEEE, Dec. 2022, pp. 1–6. doi:10.1109/ICDSAAI55433.2022.10028860.

[15] T. Wang, F. Wu, and R. O. Sinnott, "A Case Study in X Bot Identification: Are They Still a Problem?," in *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, IEEE, Dec. 2020, pp. 1–8. doi:10.1109/SNAMS52053.2020.9336537.

[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*, IEEE, Dec. 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.

[17] E. Marcelli, T. Barbariol, V. Savarino, A. Beghi, and G. A. Susto, "A Revised Isolation Forest procedure for Anomaly Detection with High Number of Data Points," in *2022 IEEE 23rd Latin American Test Symposium (LATS)*, IEEE, Sep. 2022, pp. 1–5. doi:10.1109/LATS57337.2022.9936964.

[18] C. Melquiades and F. B. de Lima Neto, "Isolation Forest-based semi-supervised Anomaly Detection of multiple classes," in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Jun. 2022, pp. 1–6. doi: 10.23919/CISTI54924.2022.9820467.

[19] M. Badurowicz, P. Karczmarek, and J. Montusiewicz, "Fuzzy Extensions of Isolation Forests for Road Anomaly Detection," in *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, Jul. 2021, pp. 1–6. doi: 10.1109/FUZZ45933.2021.9494469.

[20] J. J. Michael and M. Thenmozhi, "Outlier detection in maize field using Isolation Forest: A one-class classifier," in *2023 International Conference on Networking and Communications (ICNWC)*, IEEE, Apr. 2023, pp. 1–6. doi: 10.1109/ICNWC57852.2023.10127404.

[21] A. Petkovski and V. Shehu, "Anomaly Detection on Univariate Sensing Time Series Data for Smart Aquaculture Using K-Means, Isolation Forest, and Local Outlier Factor," in *2023 12th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/MECO58584.2023.10154991.

[22] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended Isolation Forest," *IEEE Trans Knowl Data Eng*, vol. 33, no. 4, pp. 1479–1489, Apr. 2021, doi: 10.1109/TKDE.2019.2947676.

[23] L. Zhang and L. Liu, "Data Anomaly Detection Based on Isolation Forest Algorithm," in *2022 International Conference on Computation, Big-Data and Engineering (ICCBE)*, IEEE, May 2022, pp. 87–89. doi:10.1109/ICCBE56101.2022.9888169.

[24] Y. Hara, Y. Fukuyama, K. Murakami, T. Iizaka, and T. Matsui, "Fault Detection of Hydroelectric Generators using Isolation Forest," in *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, IEEE, Sep. 2020, pp. 864–869. doi:10.23919/SICE48898.2020.9240331.

[25] Z. Yang *et al.*, "User Log Anomaly Detection System Based on Isolation Forest," in *2023 2nd International Joint Conference on Information and Communication Engineering (JCICE)*, IEEE, May 2023, pp. 79–84. doi: 10.1109/JCICE59059.2023.00025.

[26] D. Martín-Gutiérrez, G. Hernández-Peñaloza, A. Belmonte-Hernández, A. Lozano-Diez, and F. Álvarez, "A Deep Learning Approach for Robust Detection of Bots in Twitter Using Transformers," *IEEE Access*, vol. 9, pp. 54591–54601, 2021. doi:10.1109/ACCESS.2021.3068659

[27] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "BotOrNot: A System to Evaluate Social Bots," in *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 273–274, International World Wide Web Conferences Steering Committee, 2016. doi: 10.1145/2872518.2889302.

[28] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, "Online Human-Bot Interactions: Detection, Estimation, and Characterization," *arXiv:1703.03107*, Mar. 2017. doi:10.48550/arXiv.1703.03107.

[29] J. Rodríguez-Ruiz, J. I. Mata-Sánchez, R. Monroy, O. Loyola-González, and A. López-Cuevas, "A one-class classification approach for bot detection on X," *Comput Secur*, vol. 91, p. 101715, Apr. 2020, doi: 10.1016/j.cose.2020.101715.

[30] Y. Chabchoub, M. U. Togbe, A. Boly, and R. Chiky, "An In-Depth Study and Improvement of Isolation Forest," *IEEE Access*, vol. 10, pp. 10219–10237, 2022, doi: 10.1109/ACCESS.2022.3144425.

[31] S. Liu, Z. Ji, and Y. Wang, "Improving Anomaly Detection Fusion Method of Rotating Machinery Based on ANN and Isolation Forest," in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, IEEE, Jul. 2020, pp. 581–584. doi:10.1109/CVIDL51233.2020.00-23.

[32] J. Su and J. Li, "An Anomaly Detection Algorithm for Multi-dimensional Segmentation Plane Isolation Forest," in *2022 IEEE 5th International Conference on Computer and Communication Engineering Technology (CCET)*, IEEE, Aug. 2022, pp. 89–93. doi:10.1109/CCET55412.2022.9906369.

[33] R. ELHadad, Y.-F. Tan, and W.-N. Tan, "Comparison of Enhanced Isolation Forest and Enhanced Local Outlier Factor in Anomalous Power Consumption Labelling," in *2023 IEEE 3rd International Conference in Power Engineering Applications (ICPEA)*, IEEE, Mar. 2023, pp. 243–247. doi: 10.1109/ICPEA56918.2023.10093186.

[34] P. Yu and L. Jia, "Wind Power Data Cleaning Based on Autoencoder-Isolation Forest," in *2022 7th International Conference on Power and Renewable Energy (ICPRE)*, IEEE, Sep. 2022, pp. 803–808. doi:10.1109/ICPRE55555.2022.9960342.