Bot Detection on TwiBot-22: A Hybrid Approach Using Structured Metadata and Linguistic Features

Massimiliano Capparuccia[1*]

[1*]Department of Computer Science "Giovanni degli Antoni", University of Milan, Milan, Italy.

Corresponding author(s). E-mail(s):
massimiliano.capparuccia@studenti.unimi.it;

Abstract

This project investigates whether language and profile-activity signals are sufficient to flag AI-generated or automated personas on social media. We operationalize the task as binary classification (*bot* vs. *human*) and train a compact gradient-boosted decision tree model on a blend of textual and structured cues. On TwiBot-22, using sentence-transformer embeddings of user bios and tweets together with profile/timeline metadata, our full model achieves a test accuracy of 70.3% and a bot F1-score of 57.8%.

This result significantly outperforms a structural-only baseline (F1: 53.8%) and a variant using only bio text (F1: 55.0%). The findings demonstrate that while structured features provide a strong foundation, linguistic content from the user's timeline is a critical component for improving detection. Feature importance analysis confirms that the model relies on a mix of metadata and text signals, suggesting that this lightweight, graph-agnostic approach is effective for identifying automated footprints based on both activity and language.

Keywords: Social Bot Detection, TwiBot-22, Feature Importance, Text Embeddings, Structured Metadata

1 Introduction

The proliferation of fake news and fake profiles on social media platforms poses significant threats to information integrity and societal trust. These profiles are often used in Advanced Persistent Threat cases to spread malware or links to it.[1] Beyond the classical notion of social bots driven by rule-based schedulers, we now face accounts whose

surface text is linguistically polished, while their overall activity remains mechanically orchestrated. This work explores whether a lightweight model, combining transformer-based text embeddings with profile and activity descriptors, can detect such accounts using only public signals.

Research question. *To what extent can language cues and simple multimodal footprint features discriminate bots from humans in a contemporary Twitter/X dataset?*

2 Related Work

Early efforts to detect social media bots often relied on simple content and behavior heuristics. Accounts that repeatedly posted duplicate or formulaic messages could be caught with rule-based filters (e.g. capping duplicate tweets). Classical approaches identified spam bots through features like spammy keywords, high posting rates, or excessive self-repetition [2]. These methods proved effective against first-generation bots that had unnatural consistency in content. However, the "arms race" nature of this domain soon became evident: as detection improved, bots evolved in sophistication. A broad array of supervised classifiers were developed in the 2010s, especially within the NLP community, to analyze an account's textual footprint (profile descriptions, posts) and predict if it was human or bot. [3].

3 Data

We use the **TwiBot-22** [4] corpus (user profiles, timelines, and labels). [1] Users and tweets are streamed from JSON with `ijson` to limit memory usage. For each user we aggregate: account age (days), engagement averages (likes/retweets/replies/quotes), activity counts, and binary flags (`protected`, `default_profile_image`, `verified` if present). For text, we build a document per user from the profile *bio* and the last N tweets (after lightweight cleaning and near-duplicate filtering). We adopt TwiBot-22 as our primary dataset because, to the best of our knowledge, it is the largest and most comprehensive benchmark for Twitter bot detection currently available, offering a heterogeneous graph with users, tweets, lists, and hashtags.

3.1 Feature Selection

The selection of the feature keys employed from the TwiBot-22 dataset in this study was informed by the analytical framework described in [5]. The comprehensive explanation provided in that work offered valuable insights into the dataset's structure and the interpretative role of its attributes. Following this guidance, we identified and selected the keys most relevant to the objectives of the present research, specifically those contributing to the characterization of user behavior and interaction dynamics within Twitter data. The final set of features includes for tweets author_id and text, as well as engagement, related metrics such as like_count, retweet_count, reply_count,

[1] As described in the accompanying repository. Labels are resolved prioritizing: an in-file `label` column in `user.json`; otherwise `label.json` or `ground_truth.json`.

and quote_count. For user-level information, the chosen attributes comprise followers_count, following_count, listed_count, tweet_count, account_age, and various profile flags indicating account characteristics. This combination of tweet- and user-based features ensures methodological consistency with the referenced analysis and provides a focused yet comprehensive subset of attributes suitable for the detection and classification tasks addressed in this study.

4 Method

4.1 Text Encoder

We encode texts with the sentence-transformer `paraphrase-multilingual-MiniLM-L12-v2` (384 d). Embeddings are computed with mean pooling over token representations (cosine-normalized). We obtain a 384-d vector for the bio and a 384-d vector for the average of a user's tweets.

4.2 Structured Features

Structured features include: followers/following/listed counts, total tweets, account age, profile flags, per-user averages of likes/retweets/replies/quotes, number of tweets N (we use N=20), and a simple *unique_text_ratio*.

4.3 Classifier and Training Protocol

We concatenate *(structured features* $\|$ *bio embedding* $\|$ *tweet embedding)* and train an **XGBoost** classifier with early stopping on a validation split. Class imbalance is mitigated with `scale_pos_weight` computed from the training set. Standardization is applied to numeric features where appropriate.

4.4 Ablation protocol.

To assess the relevance of textual sources, we train three variants: **S** (Structural only), **B** (Bio + Structural, no tweets), and **F** (Full: Bio + Tweets + Structural). Table 1 reports the test performance of each variant.

4.5 Validation and Test

We adhere to the official TwiBot-22 dataset split, using the provided `train` (700k users), `val` (200k users), and `test` (100k users) sets. The XGBoost model was trained on the `train` set, using the `val` set for early stopping (`eval_set`) to prevent overfitting. As the TwiBot-22 test set is imbalanced (29.4% bot class), standard 0.5 thresholding is suboptimal for maximizing the F1-score. We therefore determined an optimal classification threshold by evaluating the F1-score (for the bot class) across the `val` set's prediction probabilities. This process yielded an optimal threshold of 0.1557, which was used for all final predictions on the `test` set.

5 Results

The performance of the full model (F), combining structured features, bio embeddings, and tweet embeddings, is presented in Figures 1 through 3. The model achieved a test accuracy of **70.3%** and a bot-class F1-score of **57.8%**. The Area Under the ROC Curve (AUC-ROC) was **0.770**, indicating good discriminative ability (Fig 1). The Precision-Recall (PR) curve (Fig 2) shows the trade-off, yielding an Average Precision (AUCPR) of 0.601.

Ablation Study.

To understand the contribution of each modality, we compare the full model (F) against the structural-only (S) and bio-plus-structural (B) variants. Results are summarized in Table 1. The structural-only model (S) achieves a baseline F1-score of 53.8%. Adding the 384-d bio embedding (Model B) provides a modest performance lift, increasing F1 to 55.0% and AUC-ROC to 0.755. The most significant improvement comes from adding the 384-d average tweet embedding (Model F), which boosts the F1-score by nearly 3 points to 57.8% and accuracy by over 6 points compared to the bio-only model.

This ablation study clearly demonstrates that while structured features provide a strong baseline, textual content is crucial for improved performance. Notably, the aggregated signal from an account's *timeline* (tweets) is a much stronger indicator than its *static bio*.

Table 1 Ablation study results on the TwiBot-22 test set. (F) = Full, (B) = Bio + Structural, (S) = Structural only.

Model	Accuracy	F1 (Bot)	AUC-ROC	Bal. Acc.
(S) Struct only	0.6012	0.5380	0.7244	0.6558
(B) Bio + Struct	0.6375	0.5499	0.7554	0.6709
(F) Full	**0.7029**	**0.5783**	**0.7702**	**0.6997**

Comparison with TwiBot-22 Official Baselines

We compare our full multimodal model (F) against the baselines published in the official TwiBot-22 repository [4]. The results are summarized in Table 2.

Our approach (Accuracy: 0.703 , F1: 0.578), which utilizes only user-level structured features and text embeddings without any graph data, successfully outperforms all official *feature-based* baselines. It shows a significant improvement over the standard GBDT (Accuracy: 0.648, F1: 0.570) and Random Forest (Accuracy: 0.642, F1: 0.563) models, demonstrating the strong predictive value added by the MiniLM text embeddings.

When compared to the *language-based* baseline (RoBERTa), our lightweight XGBoost + MiniLM model achieves a higher accuracy (0.703 vs. 0.684) but a lower bot F1-score (0.578 vs. 0.635). This suggests our multimodal blend is highly effective at correctly

Table 2 Comparison with official TwiBot-22 baselines. Accuracies and F1 for baselines are taken from the official repository [4].

Category	Method	Graph	Accuracy	F1 (bot)
Feature-based	SVM	No	0.635	0.551
	Random Forest (RF)	No	0.642	0.563
	GBDT	No	0.648	0.570
Language-based	RoBERTa	No	0.684	0.635
GNN-based	BotRGCN	Yes	0.812	0.796
	TwiBot-22 (Full)	Yes	0.849	0.838
Ours (Full)	**XGB+MiniLM (F)**	**No**	**0.703**	**0.578**

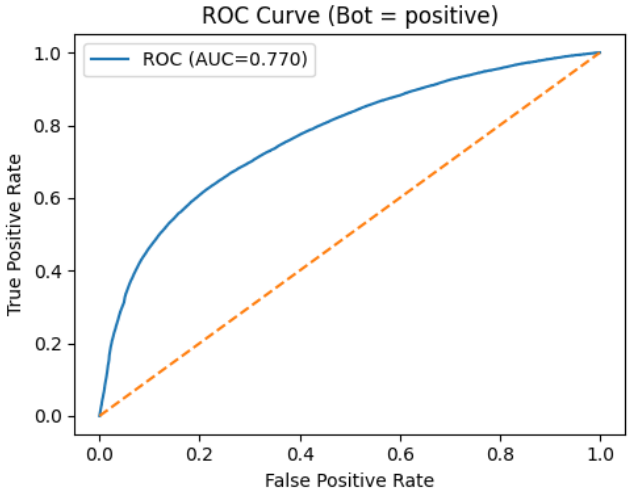

Fig. 1 ROC curve on the test split.

identifying the majority class (humans), while the fine-tuned RoBERTa model is better optimized for minority class (bot) recall.

As expected, our model does not compete with the state-of-the-art *GNN-based* methods (e.g., BotRGCN: 0.812 Acc, 0.796 F1; TwiBot-22 Full: 0.849 Acc, 0.838 F1) which leverage the rich relational graph structure unavailable to our model. However, our findings confirm that strong detection performance can be achieved using only local multimodal footprints, outperforming traditional feature-only approaches.

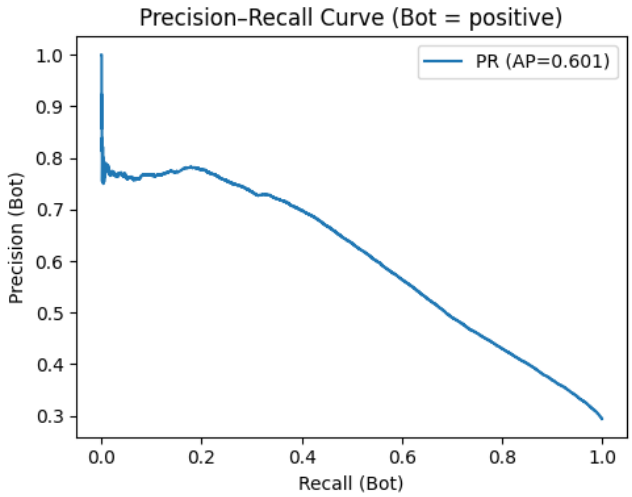

Fig. 2 Precision–Recall curve

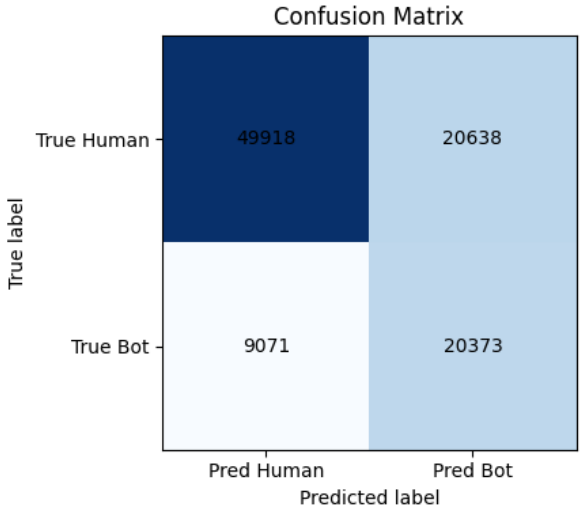

Fig. 3 Confusion matrix

6 Analysis and Discussion

What signals the model used.

To understand what signals the model learned to prioritize, we analyze the XGBoost feature importance (measured by 'gain'), as shown in Figure 4. The analysis reveals that the model relies overwhelmingly on high-level structured metadata. The `pm.followers_count` feature is dominant, providing more than three times the gain of any other signal. This suggests the model identifies bots primarily by their (lack

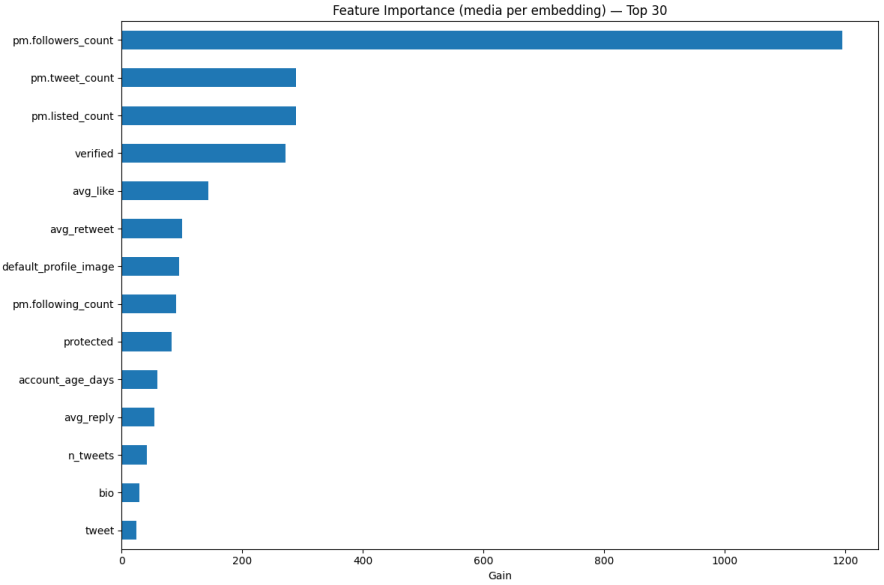

Fig. 4 Feature importance (XGBoost gain)

of) popularity. A second tier of important features consists of other metadata, such as `pm.tweet_count`, `pm.listed_count`, and `verified`.

A crucial observation arises when comparing this importance ranking (Fig 4) with the ablation study (Table 1). The linguistic embeddings (`bio` and `tweet`) are ranked as the two least important features by gain. However, the ablation study clearly shows that removing them significantly degrades performance (e.g., the F1-score drops from 0.578 to 0.550 when removing tweets).

This is not a contradiction. It indicates that while the model builds its main structure on high-gain metadata (like follower counts), these linguistic features provide unique, complementary information that metadata alone cannot capture. The model likely uses these "low-gain" text features in deeper, more specific splits to resolve ambiguity and correct errors left by the high-level features. Therefore, while metadata features are the most important signal, the linguistic features are necessary to achieve the model's best performance.

Weaknesses and threats to validity.

The model's performance, while promising, highlights several weaknesses. The final F1-score of 57.8% indicates significant room for improvement. The classification report and confusion matrix (Fig 3) reveal the primary challenge: while human recall is high (0.71), the bot recall is only 0.69. This means that even with an optimized threshold, the model misses 31% of bots. Furthermore, the model relies heavily on metadata (Fig 4). This makes it vulnerable to "cheap fakes" that can easily mimic human-like follower/following ratios or acquire verification, while its ability to detect sophisticated *linguistic* fakes (where text embeddings would dominate) appears limited.

7 Reproducibility

The repository contains the full pipeline: streaming JSON loading, label resolution, feature aggregation, MiniLM embedding, XGBoost training with early stopping, and evaluation/plots. To ease replication we cache embeddings to Parquet/CSV and fix random seeds. All experiments for this project were conducted on a 2021 MacBook Pro equipped with 16 GB of RAM, which demonstrates that the proposed pipeline can be executed on widely available consumer hardware without the need for dedicated GPU resources.

8 Conclusion

This project demonstrated the feasibility of detecting fake personas on TwiBot-22 using a lightweight XGBoost model trained on multimodal footprints. By combining structured metadata with transformer-based embeddings of user bios and timelines, our full model achieved a test accuracy of 70.3% and a bot F1-score of 57.8%. An ablation study revealed that while structured features provide a strong baseline (53.8% F1), linguistic content offers a significant performance boost. Notably, aggregated tweet content proved more valuable (57.8% F1) than static bio text (55.0% F1). However, feature importance analysis showed that the model relies heavily on high-level metadata like verification status and follower counts, suggesting vulnerability to more sophisticated fakes. Future work should incorporate graph-based features (follower/friend networks) and more advanced linguistic models to capture the subtle inconsistencies of AI-generated text, moving beyond simple embeddings.

Data and Code Availability. Scripts for parsing TwiBot-22 and reproducing the ablations are provided at: `https://github.com/MaxKappa/botornot`

Data and Code Availability.

References

[1] E. Papageorgiou, C. Chronis, I. Varlamis, Y. Himeur, A survey on the use of large language models (llms) in fake news. Future Internet **16**(8) (2024). https://doi.org/10.3390/fi16080298. URL https://www.mdpi.com/1999-5903/16/8/298

[2] R. Veit, M. Lones. A comparative analysis of transformer models in social bot detection (2025). URL https://arxiv.org/abs/2509.14936

[3] S. Cresci, A decade of social bot detection. Communications of the ACM **63**(10), 72–83 (2020). https://doi.org/10.1145/3409116. URL http://dx.doi.org/10.1145/3409116

[4] S. Feng, Z. Tan, H. Wan, N. Wang, Z. Chen, B. Zhang, Q. Zheng, W. Zhang, Z. Lei, S. Yang, et al., *TwiBot-22: Towards Graph-Based Twitter Bot Detection*, in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2022)

[5] I. Sakhabutdinov. Bot detection in social media: An empirical study using twibot-22 (2025). URL https://www.diva-portal.org/smash/get/diva2:1984105/FULLTEXT01.pdf