A Comprehensive Review of Machine Learning Approaches for Bot Detection on Social Media

Karlo Papa
kpapa@oxy.edu
Occidental College

1 Prior Work

Bot detection on social media platforms has been extensively studied, with certain methodologies emerging as standard approaches due to their effectiveness, interpretability, and ease of implementation. However, these solutions are not without limitations, prompting the exploration of alternative methods and hybrid approaches.

1.1 Supervised Learning as the Standard Approach

Supervised learning is the dominant method of bot detection. These models rely on labeled datasets to train classifiers that distinguish between bots and human accounts. Among the most commonly used algorithms are Random Forests and Support Vector Machines (SVMs), both of which are favored for their ability to handle high-dimensional data [7]. Logistic Regression has also been adopted due to its simplicity and interpretability, though its ability to model complex patterns is limited [3]. The popularity of supervised learning methods can be attributed to their well-understood theoretical foundations and strong performance when clear distinctions exist between bots and human users. Additionally, these models have been extensively validated across various studies, solidifying their status as the default approach in bot detection research [7, 6]. The most significant limitation of supervised learning methods is their reliance on high-quality labeled data, which is both expensive and time-consuming to obtain. Furthermore, bot behaviors continuously evolve, meaning that models trained on historical data may fail to generalize to emerging threats. Supervised classifiers also struggle with adversarial bots designed to mimic human behavior, reducing their effectiveness over time [11].

1.2 Exploring Unsupervised Learning

To address the limitations of labeled data dependency, researchers have explored unsupervised learning techniques. Clustering algorithms, such as K-means, have been employed to group accounts with similar behavioral characteristics, revealing bot networks [5]. Similarly, anomaly detection methods like Isolation Forests have been used to flag outliers, which may correspond to automated activity [4]. Clustering approaches can group similar accounts together but may struggle to distinguish between sophisticated bots and legitimate but atypical human users. Anomaly detection techniques, while effective in highlighting unusual activity, often produce high false positive rates, incorrectly flagging real users as bots. This trade-off between sensitivity and precision remains a key issue in unsupervised bot detection.

1.3 Deep Learning and Its Challenges

Deep learning has introduced more powerful methods for bot detection, leveraging neural networks to uncover intricate patterns in user behavior and textual data. Long Short-Term Memory (LSTM) networks have proven effective in capturing temporal dependencies, making them well-suited for analyzing tweet sequences [8]. Convolutional Neural Networks (CNNs) have been applied to text classification, treating sequences of words or characters as structured data. Despite their high performance, deep learning models require extensive computational resources and large labeled datasets for training, which are often unavailable. Additionally, deep learning models are frequently criticized for their lack of interpretability, making them less desirable in contexts where transparency is important [1].

1.4 Drawing Inspiration from Related Fields

Bot detection shares similarities with problems in other domains, particularly cybersecurity and NLP. In cybersecurity, anomaly detection techniques have been successfully used to identify network intrusions and fraudulent activity. Similarly, in NLP, sentiment analysis and text classification techniques have been adapted for bot detection by analyzing linguistic features and posting patterns [5]. Another promising direction is the integration of hybrid approaches that combine multiple methodologies to compensate for their respective weaknesses. For example, combining supervised learning with anomaly detection can enhance generalizability while reducing reliance on labeled data [6]. Addition-

ally, feature engineering techniques inspired by social media dynamics—such as analyzing user engagement metrics, network structures, and metadata—can further improve detection accuracy.

2 Technical Background

Detecting automated accounts (bots) on social media platforms like Twitter can be tackled using a variety of machine learning techniques. The most common methods fall into three broad categories: supervised learning, unsupervised learning, and deep learning. Each approach has strengths and limitations depending on the nature of the data, the complexity of bot behavior, and computational constraints. Furthermore, effective bot detection requires a strong understanding of social media dynamics, cybersecurity, and linguistics.

2.1 Supervised Learning Approaches

Supervised learning relies on labeled datasets to classify accounts as bots or humans. Popular algorithms include Random Forest, Support Vector Machines (SVM), and Logistic Regression. Random Forest uses ensemble learning to construct multiple decision trees and aggregates their predictions, making it robust to overfitting and suitable for high-dimensional datasets [4]. SVMs identify an optimal hyperplane to separate data classes, making them particularly effective for binary classification tasks [6]. Logistic Regression, assumes a linear decision boundary and may struggle to capture complex bot behaviors [3]. Supervised learning models can achieve high accuracy when trained on quality data but suffer from a variety of limitations. They require extensive labeled datasets, bot labeling is subjective, and behavioral patterns evolve over time, meaning models must be frequently retrained to remain effective [11].

2.2 Unsupervised Learning Approaches

Unsupervised learning methods identify patterns or anomalies without labeled training data. One common approach uses K-means clustering to group similar accounts based on behavioral similarities, potentially exposing bot networks. Another method is anomaly detection, which flags outliers that exhibit suspicious activity. These techniques are particularly useful when labeled data is scarce or when detecting previously unseen bot behaviors. However, validation remains a challenge, as there is no labeled ground truth to confirm whether an identified cluster or outlier truly represents bot activity. Additionally, unsupervised models may generate false positives by misclassifying legitimate but atypical users [14].

2.3 Deep Learning Approaches

Deep learning models, particularly Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), provide more advanced methods for bot detection. LSTM networks are well-suited for capturing temporal dependencies in sequential data, making them effective for analyzing tweet sequences and user behavior over time [3]. CNNs can be applied to text classification by treating words or characters as structured sequences [2]. While deep learning approaches excel at detecting complex bot behaviors, they require large training datasets and significant computational resources. Additionally, these models often function as black boxes, making it difficult to interpret their decision-making processes. Given these limitations, deep learning is most effective in large-scale bot detection where computational power is available and model explainability is less critical [1].

2.4 Other Domains

Beyond technical considerations, bot detection also requires expertise from other domains. Understanding social media dynamics is crucial, as bot behavior often mimics real user engagement [15] [8]. Cybersecurity knowledge helps in recognizing adversarial techniques used to bypass detection, such as coordinated bot networks and deceptive interactions [10]. Linguistics plays a vital role in applying natural language processing (NLP) techniques to analyze tweet content, detect unnatural text patterns, and identify automated responses.

2.5 Relevance and Impact

This research has significant implications for multiple stakeholders. For researchers, improving bot detection contributes to advancements in machine learning and cybersecurity [9]. Social media platforms benefit from enhanced detection methods that help maintain the integrity of online discourse by reducing misinformation and spam [6]. For the broader public, better bot detection raises awareness of automated influence campaigns and their potential effects on public opinion and security [5]. By integrating machine learning with insights from social sciences and cybersecurity, this work aims to enhance the accuracy and robustness of bot detection on social media.

3 Methods

This paper proposes a hybrid approach that integrates supervised learning and deep learning techniques to improve bot detection accuracy. While traditional classifiers offer interpretability and efficiency, deep learning models excel

at capturing complex patterns. Combining these techniques aims to balance precision, adaptability, and computational feasibility.

3.1 Framework and Algorithm Selection

Supervised learning has been widely used in bot detection, with Random Forests and Support Vector Machines (SVMs) models forming the foundation for analyzing metadata and behavioral patterns, such as posting frequency, account age, and follower ratios. These supervised methods can be enhanced by deep learning approaches such as Long Short-Term Memory (LSTM) networks that have proven effective in capturing more nuanced bot behaviors by modeling temporal dependencies to identify automation. Convolutional Neural Networks (CNNs) can extract linguistic and stylistic features from tweet content, enabling the detection of unnatural text patterns indicative of automated accounts [2]. By applying CNNs to textual features and LSTMs to sequential data, we enhance the model's ability to distinguish bots from human users. This ensemble approach leverages the interpretability of supervised learning while incorporating the adaptive strengths of deep learning, making it more robust against sophisticated bots that evade traditional classifiers [6].

3.2 Hyperparameter Selection and Optimization

This section explores a variety hyperparameters that will influence classification accuracy, generalization, and computational efficiency. For Random Forests, the number of trees (n_estimators) should start between 100 and 500, increasing the number of trees should accuracy but also increases computational cost. Additionally, the maximum tree depth (max_depth) should be between 10 and 50 to balance complexity and avoid overfitting, since deeper trees can capture more intricate patterns but may also learn noise in the data. The SVMs will experiment with linear, polynomial, and radial basis function (RBF) kernels. Linear kernels are computationally efficient and work well when data is largely separable, while RBF kernels can capture non-linear decision boundaries, which may better represent bot behaviors. I'll also adjust the regularization parameter (C), testing values between 0.1 and 10 to adjust the trade-off between maximizing the margin and minimizing misclassification errors. The LSTMs will have between 50 and 200 hidden units per layer to evaluate how much sequential information the network retains. A higher number of units allows for richer representations of tweet sequences but increases the risk of overfitting. To address overfitting, I'll experiment with dropout rates between 0.2 and 0.5, randomly deactivating neurons during training to improve generalization [13]. CNNs will vary the number of filters between 32 and 128 to determine how many feature detectors should be applied at each convolutional layer. More filters enable the extraction of richer text features but also increase computational demands. Additionally, I'll test filter sizes of 3, 5, and 7, which determine the number of consecutive words or characters analyzed together. Smaller filters capture fine-grained word-level details, while larger filters identify broader semantic patterns, both of which are valuable for detecting bots using varied linguistic styles [13].

3.3 Implementation and Evaluation Strategy

To ensure reliability, I'll train and evaluate my models on publicly available datasets containing labeled bot and human accounts. Performance will be assessed using standard classification metrics (explored in the next section), with particular emphasis on minimizing false positives, which are a major concern in bot detection [12]. Additionally, ablation studies could be conducted to isolate the impact of different model components and determine their relative contributions to overall detection accuracy.

4 Metrics and Results

Evaluating the success of bot detection models requires a range of performance metrics to assess accuracy, robustness, and practical applicability. Previous studies have primarily relied on classification-based metrics such as accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) to measure the effectiveness of their models. This section reviews commonly used evaluation criteria in prior research along with their typical success thresholds.

4.1 Evaluation Metrics in Prior Work

Accuracy measures the proportion of correctly classified instances, but it can be misleading in imbalanced datasets where human accounts vastly outnumber bots [7]. A model with high accuracy may still perform poorly in detecting bots if it disproportionately classifies instances as human. To address this limitation, precision and recall have been widely used. Precision quantifies the proportion of correctly identified bots among all accounts flagged as bots, ensuring that false positives are minimized [6]. Recall, in contrast, measures the proportion of actual bots that the model successfully detects, highlighting its effectiveness in reducing false negatives. The F1-score, the harmonic mean of precision and recall, provides a balanced assessment when working with imbalanced datasets and is commonly used in bot detection studies [3]. AUC-ROC is another critical metric that evaluates the trade-off between true positive and false positive rates. The AUC score reflects the likelihood that

a randomly selected bot will be ranked higher than a randomly selected human account. Models with an AUC score close to 1 are generally considered well-calibrated, as they effectively differentiate between bots and legitimate users [5]. These classification-based metrics form the foundation for evaluating bot detection models and have been widely adopted in prior research.

4.2 Thresholds for Success in Prior Studies

The thresholds that constitute a "successful" bot detection model vary depending on the study's objectives and constraints. Many studies have reported accuracy above 90% as a benchmark for strong performance [2]. However, given the trade-offs between different types of errors, precision and recall values above 85% are often considered more indicative of a model's reliability in real-world applications [4]. AUC-ROC values above 0.9 are typically seen as strong indicators that a model is effectively distinguishing between classes [11]. In some cases, studies prioritize precision over recall to minimize false positives, particularly when misclassifying legitimate users could have negative consequences. Conversely, applications that aim to detect coordinated bot networks may prioritize recall to ensure that as many bots as possible are identified, even at the cost of increased false positives.

4.3 Additional Metrics for Evaluation

Beyond traditional classification metrics, additional measures can provide deeper insights into model performance. The Matthews Correlation Coefficient (MCC) is one such metric, offering a more balanced evaluation of classification quality by considering true positives, false positives, true negatives, and false negatives in a single calculation. Unlike accuracy, MCC is particularly useful when class distributions are imbalanced, making it a valuable alternative for bot detection. Computational efficiency is another key consideration, as real-world bot detection systems must process large volumes of data in real time. Evaluating model inference speed and resource consumption helps determine whether an approach is feasible for deployment on social media platforms [2]. While prior studies have primarily focused on classification accuracy, integrating efficiency metrics ensures that models remain practical at scale.

References

[1] Adams, Terrence. "AI-Powered Social Bots". In: *arXiv* (2017). URL: https://arxiv.org/abs/1706.05143.

[2] Akyon, Fatih Cagatay and Kalfaoglu, Esat. "Instagram Fake and Automated Account Detection". In: *IEEE*. 2019. DOI: 10.1109/ICCVW.2019.00123. URL: https://www.researchgate.net/publication/338364922_Instagram_Fake_and_Automated_Account_Detection.

[3] Cai, Chiyu, Li, Linjing, and Zeng, Daniel. "Behavior Enhanced Deep Bot Detection in Social Media". In: *ISI*. 2017. DOI: 10.1109/ISI.2017.8004887. URL: https://www.researchgate.net/publication/319054413.

[4] Grimme, Christian, Assenmacher, Dennis, and Adam, Lena. "Changing Perspectives: Is It Sufficient to Detect Social Bots?" In: *SCSM*. 2018. DOI: 10.1007/978-3-319-91521-0_32. URL: https://link.springer.com/chapter/10.1007/978-3-319-91521-0_32.

[5] Hajli, Nick et al. "Social Bots and the Spread of Disinformation in Social Media: The Challenges of Artificial Intelligence". In: *British Journal of Management* (2021). DOI: 10.1111/1467-8551.12554. URL: https://onlinelibrary.wiley.com/doi/10.1111/1467-8551.12554.

[6] Hayawi, Kadhim et al. "Social media bot detection with deep learning methods: a systematic review". In: *Neural Computing and Applications* (2023). DOI: 10.1007/s00521-023-08352-z. URL: https://doi.org/10.1007/s00521-023-08352-z.

[7] Heidari, Maryam, Jones, James H Jr, and Uzuner, Ozlem. "An Empirical Study of Machine learning Algorithms for Social Media Bot Detection". In: *IEMTRONICS*. 2021. DOI: 10.1109/IEMTRONICS52119.2021.9422605. URL: https://www.researchgate.net/publication/351596484.

[8] Kenny, Ryan et al. "Duped by Bots: Why Some are Better than Others at Detecting Fake Social Media Personas". In: *Human Factors* (2022). DOI: 10.1177/00187208211072642. URL: https://journals.sagepub.com/doi/10.1177/00187208211072642.

[9] Kudugunta, Sneha and Ferrara, Emilio. "Deep Neural Networks for Bot Detection". In: *arXiv* (2018). URL: https://arxiv.org/abs/1802.04289.

[10] Mbona, Innocent and Eloff, Jan H.P. "Classifying social media bots as malicious or benign using semi-supervised machine learning". In: *Journal of Cybersecurity* (2023). DOI: 10.1093/cybsec/tyac015. URL: https://doi.org/10.1093/cybsec/tyac015.

[11] Orabi, Mariam et al. "Detection of Bots in Social Media: A Systematic Review". In: *Information Processing & Management* (2020). DOI: 10.1016/j.ipm.2020.102250. URL: https://doi.org/10.1016/j.ipm.2020.102250.

[12] Rauchfleisch, Adrian and Kaiser, Jonas. "The False positive problem of automatic bot detection in social science research". In: *PLOS ONE* (2020). DOI: 10.1371/journal.pone.0241045. URL: https://doi.org/10.1371/journal.pone.0241045.

[13] Wei, Feng and Nguyen, Uyen Trang. "Twitter Bot Detection Using Bidirectional Long Short-term Memory Neural Networks and Word Embeddings". In: *arXiv* (2020). URL: https://arxiv.org/abs/2002.01336.

[14] Yang, Kai-Cheng and Menczer, Filippo. "Anatomy of an AI-powered malicious social botnet". In: *arXiv* (2023). URL: https://arxiv.org/abs/2307.16336.

[15] Yu, Jingru et al. "The Shadow of Fraud: The Emerging Danger of AI-powered Social Engineering and its Possible Cure". In: *arXiv* (2024). URL: https://arxiv.org/abs/2407.15912.