People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 Mai 1945 - Guelma
Faculty of Mathematics, Computer Science and Material Science
Computer Science Department

Master's thesis

Branch: Computer Science

Option: Information and Communication Sciences and Technologies

Theme:

Enhancing Social Network Security: Machine Learning-Based Bot Detection

Presented By
Zied Talha

In Front of The Jury

Dr. Abderrahmen Keffali	President
Dr. Abdelhakim Hannousse	Supervisor
Dr. Nadjette Benhamida	Reviewer

June 2024

This thesis is dedicated to my parents, whose strength and support have been my foundation. To my mother, for her joy and wisdom, teaching me resilience and determination. Your guidance shaped my journey. Thank you, Mum. To my father, a self-made man, for his dedication and hard work. Your values of perseverance and integrity are my guiding light. I am grateful for your lessons. To my brothers, Yahia and Mehdi, for their encouragement and camaraderie. To my grandparents, for their love and wisdom. To my friends, Abdelhamid, Abdenour, Rami, Louai, Oussama, Youcef, Tajou, and Billel, for their unwavering support and shared joy. Special thanks to my colleagues Safir and Youssef for the memories. Each of you has significantly shaped my life. Thank you.

Talha Zied.

Acknowledgements

I extend my heartfelt gratitude to Allah, for His blessings and for illuminating my path with opportunities throughout all my life.

My deepest appreciation goes to Dr. Abdelhakim Hannousse, my supervisor, for his guidance, invaluable advice, and support that navigated me through every stage of this project. It has been an absolute honor to work under your mentorship.

I would also like to express my appreciation to the members of the jury for their time, expertise, and valuable feedback during the evaluation of this thesis. Your insights have contributed significantly to its refinement and quality.

I want to thank all members of the Computer Science Department at 8th May University - Guelma, whose knowledge and assistance have been instrumental in shaping my understanding and skills in this field.

To my colleagues, whose collaboration and insights have enriched my research experience, I express sincere thanks.

My gratitude also extends to my family for their unwavering encouragement, understanding, and love throughout this endeavor. Your support has been my pillar of strength.

To my friends, your encouragement, camaraderie, and positivity have made this journey not just manageable but also enjoyable.

Talha Zied.

ملخص

انتشار منصات وسائل التواصل الاجتماعي قد غير من حيث التواصل، لكنه أدى أيضًا إلى ظهور الروبوتات في وسائل التواصل الاجتماعي التي يمكنها نشر الإشاعات، وتلاعب الرأي العام، والتأثير على سلامة الحوار عبر الإنترنت. تتناول هذه الأطروحة المسألة الحرجة لاكتشاف روبوتات وسائل التواصل الاجتماعي على تويتر. غالبًا ما تكون الطرق التقليدية للاكتشاف غير كافية بسبب الطبيعة المتطورة لهذه الروبوتات وكمية البيانات الضخمة المتورطة. للتغلب على هذه التحديات، تقترح هذه البحث نموذج مجموعة مختلطة يجمع بين ميزات مستندة إلى الملف الشخصي والمحتوى مع تقنيات متقدمة لمعالجة اللغة الطبيعية. تستوعب هذه الطريقة مجموعة واسعة من سلوكيات وميزات الروبوتات، مما يؤدي إلى اكتشاف أكثر دقة ومتانة. تشمل الأطروحة استعراضًا لمنصات وسائل التواصل الاجتماعي والتهديدات التي تواجهها الروبوتات، ومراجعة للطرق الحالية لاكتشاف الروبوتات، وشرحًا عميقًا للمنهجية المجمعة المختلطة المقترحة، وتقييمًا تجريبيًا لفعالية المنهجية مقارنة بالتقنيات الرائدة. تظهر النتائج تحسينات كبيرة في أداء الكشف، مدعمة جهود حماية بيئات وسائل التواصل الاجتماعي من الكيانات الآلية الضارة.

الكلمات المفتاحية:وسائل التواصل الاجتماعي، الروبوتات، الكشف، تويتر، تعلم الآلة، معالجة اللغة الطبيعية، نموذج هجين.

Résumé

La prolifération des plateformes de réseaux sociaux a transformé la communication, mais elle a également donné naissance à des bots de médias sociaux capables de propager des désinformations, de manipuler l'opinion publique et de compromettre l'intégrité du discours en ligne. Cette thèse aborde la question cruciale de la détection des bots de médias sociaux sur Twitter. Les méthodes de détection traditionnelles sont souvent insuffisantes en raison de la nature évolutive de ces bots et de la grande quantité de données impliquée. Pour surmonter ces défis, cette recherche propose un modèle hybride de groupe qui combine des caractéristiques basées sur le profil et le contenu avec des techniques avancées de traitement du langage naturel. Cette approche capture un large éventail de comportements et de caractéristiques de bot, ce qui se traduit par une détection plus précise et robuste. La thèse comprend un examen des plateformes de médias sociaux et des menaces posées par les bots, une revue des méthodes actuelles de détection des bots, une explication approfondie de la méthodologie hybride de groupe proposée, et une évaluation expérimentale de l'efficacité de la méthodologie par rapport aux techniques de pointe. Les résultats démontrent des améliorations significatives des performances de détection, soutenant les efforts visant à protéger les environnements de médias sociaux contre les entités automatisées nuisibles.

Mots-clés : Réseaux sociaux, Bots, Détection, Twitter, Apprentissage automatique, Traitement du langage naturel, Modèle hybride.

Abstract

The proliferation of social media platforms has transformed communication, but it has also given rise to social media bots that can spread misinformation, manipulate public opinion, and compromise the integrity of online discourse. This thesis addresses the critical issue of detecting social media bots on Twitter. Traditional detection methods often fall short due to the evolving nature of these bots and the vast amount of data involved. To overcome these challenges, this research proposes a hybrid ensemble model that combines profile-based and content-based features with advanced natural language processing techniques. This approach captures a wide range of bot behaviors and characteristics, resulting in more accurate and robust detection. The thesis includes an examination of social media platforms and the threats posed by bots, a review of current bot detection methods, an in-depth explanation of the proposed hybrid ensemble methodology, and an experimental evaluation of the methodology's effectiveness compared to leading techniques. The findings demonstrate significant improvements in detection performance, supporting efforts to protect social media environments from harmful automated entities.

Keywords: Social media, Bots, Detection, Twitter, Machine learning, Natural language processing, Hybrid model.

Table of Contents

Acknowledgements **i**

ملخص **ii**

Résumé **iii**

Abstract **iv**

Table of contents **v**

List of figures **viii**

List of tables **viii**

Acronyms and Abbreviations **x**

Introduction **2**

1. Social Media Platforms and Emerging Threats of Bots **4**

1.1. Versatility of Social Media Platforms 5
1.2. Growth of social media platforms 5
1.3. Impacts of social media platforms 6
1.4. Twitter 7
1.4.1. Tweets and Tweet structures 8
1.4.2. Twitter profiles 9
1.5. Social Media Bots 9
1.5.1. Spam bots 10
1.5.2. Social bots 10
1.5.3. Sybils 11
1.5.4. Cyborg bots 11
1.6. Rising threats of Social Media Bots 11
1.7. Conclusion 13

2. Social Media Bot Detection 14
2.1. Importance of Social Media Bot Detection 14
2.2. Social Media Bot Detection Methods 15
2.2.1. Profile-Based Detection Approaches 16
2.2.1.1. Supervised Approaches 17
2.2.1.2. Unsupervised Approaches 18
2.2.1.3. Semi-supervised Approaches 18
2.2.2. Content-Based Detection Approaches 19
2.2.2.1. Feature Engineering-based Approaches 19
2.2.2.2. Natural Language Processing-based Approaches 20
2.2.3. Hybrid Detection Approaches 23
2.2.4. Other Social Media Bot Detection Approaches 24
2.2.4.1. Crowdsourcing 24
2.2.4.2. Graph-Based Techniques 25
2.3. Conclusion 26

3. A Hybrid Ensemble Model for Social Media Bot Detection 27
3.1. Overview of the Proposed Methodology 28
3.2. Feature Extraction 29
3.2.1. Feature Engineering 29
3.2.1.1. Selected Profile-based Features 30
3.2.1.2. Proposed Content-based Features 33
3.2.2. Natural Language Processing (NLP) 37
3.2.2.1. Text Preprocessing 38
3.2.2.2. Text Vectorization 40
3.3. Selection of Best Models 41
3.3.1. Explored Classifiers 41
3.3.2. Hyperparameter tuning 43
3.3.3. Validation Process 44
3.4. Stacking Process 45
3.5. Conclusion 46

4. Experimentation and Results 48
4.1. Data Preparation 49
4.2. Feature Engineering based Detection 51
4.2.1. Feature Analysis 52
4.2.2. Results and Analysis 53
4.3. NLP-based Detection 56
4.4. Stacking based Detection 61
4.4.1. Comparison with State-of-the-art methods 63

4.5. Conclusion 64

Summary and conclusions **65**

References **67**

List of Figures

1.1. Number of users by social media platforms as of April 2024 [80]. 6
1.2. A tweet sample 8
1.3. A Twitter profile sample 9
1.4. Distribution of bot and human web traffic worldwide over the years [79] . . . 12

2.1. Classification of existing social media bot detection approaches. 16

3.1. Overview of the adopted methodology. 28

4.1. Distribution of tweets in the Twibot-20 dataset: (a) train set, (b) test set . . 50
4.2. Information Gain of Twitter User and Content Features. 53
4.3. Capturing the best BoW vector size for boosting the DNN performance. . . . 60

List of Tables

4.1. Distribution of Users in Twibot-20 50

4.2. Description of the Cresci2017 Dataset 51

4.3. Description of the Cresci2015 Dataset 52

4.4. Performance of models trained on engineered features: R: Recall, P: Precision, A: Accuracy, F1: F1-score 54

4.5. Grid parameters for various models used in RandomizedSearchCV 55

4.6. Parameters of the Best Pipeline for GradientBoostingClassifier 56

4.7. Performance of DL models using TF-IDF 57

4.8. Performance of DL models using BoW 58

4.9. Performance of DL models using Word2Vec and GloVe 59

4.10. Performance of fine-tuned LLMs 59

4.11. Description of the DNN Model Architecture. 61

4.12. Performance of the stacking model 61

4.13. Confusion matrices for base and meta-detectors: M1, M2, and M3 models . . 62

4.14. Comparison of various works on different datasets. 63

Acronyms and Abbreviations

BN	Bayesian Network
BERT	Bidirectional Encoder Representations from Transformers
Bi-GRU	Bidirectional Gated Recurrent Unit
Bi-LSTM	Bidirectional Long Short-Term Memory
Bi-SN-LSTM	Bidirectional Self Normalizing Long Short-Term Memory
BoW	Bag-of-Words
CBOW	Continuous Bag-of-Words
CNN	Convolutional Neural Network
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DF	Deep Forest
DNN	Deep Neural Network
DT	Decision Tree
GAN	Generative Adversarial Networks
GBM	Gradient Boosting Machine
GIF	Graphics Interchange Format
GloVe	Global Vectors for Word Representation
GPT	Generative Pre-trained Transformer
GPT-2	Generative Pre-trained Transformer 2
GRU	Gated Recurrent Unit
HC	Hierarchical Clustering
IG	Information Gain
LSTM	Long Short-Term Memory
LLM	Large Language Model
LOBO	Leave One Botnet Out
LR	Logistic Regression
MLP	Multilayer Perceptron
NB	Naive Bayes
NLP	Natural Language Processing
NN	Neural Network
OSN	Online Social Network

PCA	Principal Component Analysis
RF	Random Forest
RNN	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Transformer
SMB	Social Media Bot
SMO-C	Sequential Minimal Optimization for Classification
SMO-R	Sequential Minimal Optimization for Regression
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
t-SNE	t-distributed Stochastic Neighbor Embedding
US	United States
XGB	Extreme Gradient Boosting

Introduction

The advent of social media has revolutionized how people connect, communicate, and share information. Platforms like Twitter, Facebook, and Instagram have become integral to daily life, fostering real-time interactions across the globe [52]. However, alongside these benefits, social media has also introduced new challenges, particularly the rise of automated entities known as social media bots [1]. These bots can mimic human behavior to influence conversations, spread misinformation, and manipulate public opinion. As the presence of these bots grows, so does the need for effective detection and mitigation strategies to preserve the integrity of online discourse [38].

The primary issue addressed in this thesis is the detection of social media bots, specifically within the context of Twitter. Social media bots pose significant threats by spreading spam, conducting fraudulent activities, and disseminating misinformation at scale. These actions can undermine the trustworthiness of information on social media platforms and have real-world consequences, including impacting political processes and public health. Detecting these bots is a complex problem due to their evolving nature and the vast amount of data generated on these platforms daily. Traditional detection methods often fall short, necessitating more sophisticated approaches leveraging advancements in machine learning and data analysis [27, 76, 77]. Consequently, many studies have emerged to detect bots and reduce their threat, focusing on developing innovative techniques and tools to enhance the accuracy and efficiency of bot detection systems [10, 45, 70]. To address the challenge of detecting social media bots, this thesis proposes a hybrid ensemble model that integrates profile-based and content-based features with natural language processing techniques. This hybrid approach leverages the strengths of both methods, capturing a broader spectrum of bot behaviors and characteristics for more accurate and robust detection. By combining these techniques, the hybrid system can identify bots that might evade detection when only one method is used, offering a more comprehensive understanding of bot behavior across both structural and content dimensions.

The thesis is structured into 4 major chapters, each addressing different aspects of the research. The first chapter, *Social Media Platforms and Emerging Threats of Bots* explores the growth and impact of social media, with a particular focus on Twitter. It also highlights the increasing threat posed by bots and details the various types of social media bots along with their associated risks. The second chapter, *Social Media Bot Detection*, emphasizes the importance of detecting bots and reviews existing detection methods. It categorizes these methods into profile-based, content-based, hybrid, and other approaches. The third chapter, *A Hybrid Ensemble Model for Social Media Bot Detection*, describes the proposed methodology in detail, including feature extraction, models selection, and the stacking process. The fourth chapter, *Experimentation and Results*, outlines the experimental setup, presents the results, and provides an analysis comparing the proposed method with state-of-the-art techniques. This method has been tested on other datasets to ensure its generalizability, and it demonstrated effectiveness across these additional datasets as well. By leveraging all these concepts, this thesis aspires to make significant contributions to the field of social media bot detection, providing a robust framework to mitigate the growing threats posed by these automated entities.

Chapter 1

Social Media Platforms and Emerging Threats of Bots

In today's digital age, social media platforms have become deeply embedded in our daily lives, profoundly shaping how we communicate, share information, and interact with the world around us. Platforms like Facebook, Twitter, Instagram, and LinkedIn have evolved into powerful ecosystems, connecting individuals, businesses, and communities on a global scale. This chapter delves into the versatile nature of social media platforms, exploring their multifaceted capabilities and their ever-expanding utility in various aspects of our social, cultural, and economic landscapes. It examines the phenomenal growth of these platforms, with compelling statistics illustrating their extensive reach and impact on shaping societal narratives, fostering trends, and facilitating interpersonal interactions. Furthermore, the chapter takes a focused look at Twitter, one of the most prominent and influential social media platforms. It delves into the platform's unique features, such as tweets, profiles, and the real-time nature of communication, which have made Twitter a central hub for breaking news, public discourse, and global engagement. However, amidst the numerous benefits and opportunities provided by social media, the chapter also addresses the rising threat of malicious bots and their potential to disrupt and manipulate the digital ecosystem. It categorizes different types of malicious bots, shedding light on their intentions, capabilities, and the potential dangers they pose to the integrity of online discourse and information dissemination. Overall, this chapter aims to provide a comprehensive introduction to the world of social media platforms, their significance in our interconnected society, and the challenges posed by malicious actors exploiting these powerful tools. It sets the stage for a deeper exploration of the intricate dynamics and implications of social media bots in subsequent chapters.

1.1 Versatility of Social Media Platforms

Social media platforms, fundamentally, serve as digital environments designed to foster communication and content sharing among users. They function as dynamic hubs where individuals can create personalized profiles showcasing their interests, experiences, and expertise. These platforms offer a myriad of tools and functionalities that empower users to express themselves creatively, ranging from traditional text-based posts to multimedia-rich content like images, videos, and hyperlinks. Moreover, the interactive nature of social media enables users to engage with each other through various actions such as leaving comments, reacting with likes or dislikes, and sharing content across their networks [52]. Beyond these foundational features, modern social media platforms have evolved to incorporate advanced functionalities that enhance user experience and engagement. For instance, many platforms now integrate real-time messaging systems, enabling instant communication between individuals or groups. Additionally, live streaming capabilities have gained prominence, allowing users to broadcast live video content to their audience in real time. These live interactions foster a sense of immediacy and authenticity, strengthening the connection between content creators and their followers.

Furthermore, social media platforms often integrate with other online services and applications, creating interconnected digital ecosystems. This integration extends the reach and impact of user-generated content, as it can easily be shared across multiple platforms, reaching diverse audiences and maximizing engagement. Additionally, social media platforms often integrate with social media management tools such as Buffer, Hootsuite, and Sprout Social. These tools allow users to manage multiple social media accounts from a single platform, which can enhance the reach and impact of their content across diverse audiences. These tools enable users to schedule posts, monitor conversations, and analyze performance metrics for their social media activities, streamlining the management of their online presence. By integrating with various social media platforms, these tools provide a centralized hub for users to efficiently execute their social media strategies and maximize the impact of their content [74].

Overall, the multifaceted capabilities of social media platforms continue to shape and redefine the way individuals interact, share information, and build communities in the digital realm [55, 84].

1.2 Growth of social media platforms

Social media platforms have evolved into indispensable facets of contemporary life, wielding substantial influence as evidenced by compelling statistics. As of April 2024, the global user base surpasses a staggering 4.33 billion individuals [80], signifying over half of the world's populace actively engaged in these digital realms. Notably, Facebook commands the largest audience with approximately 3.06 billion monthly active users, closely trailed by YouTube boasting 2.50 billion, WhatsApp at 2 billion, Instagram drawing in 2 billion users, and Twitter maintaining a substantial 611 million users [53].

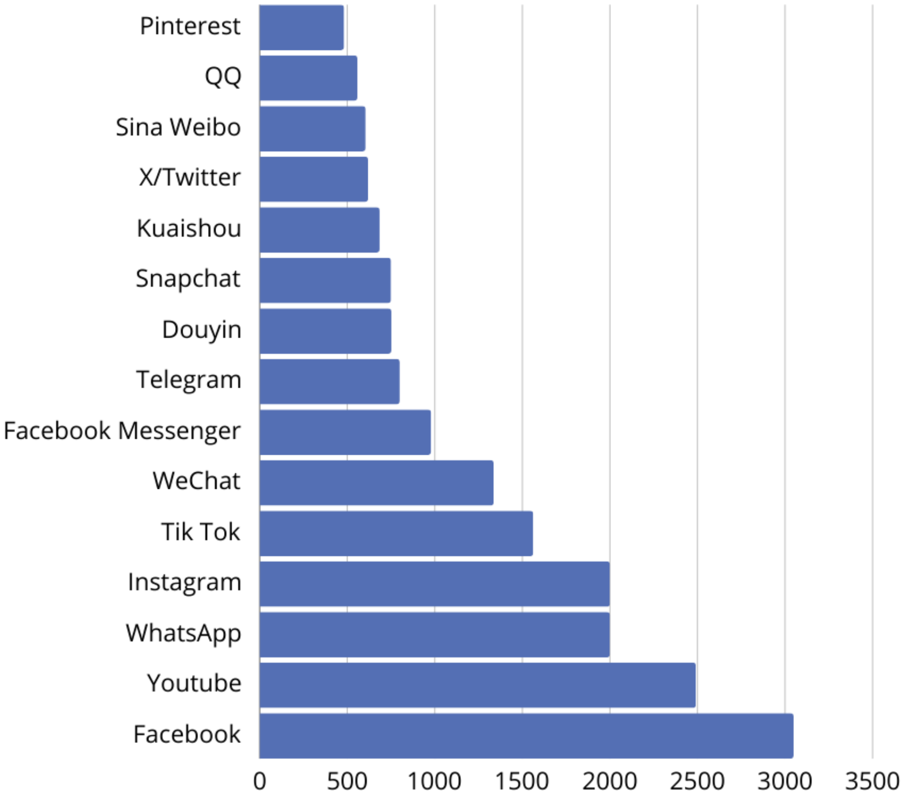


Figure 1.1 – Number of users by social media platforms as of April 2024 [80].

These numbers as shown in Figure 1.1 vividly illustrate the extensive reach and impact of social media platforms in shaping societal narratives, fostering trends, and facilitating interpersonal interactions. They have become pivotal conduits for communication, information dissemination, and cultural exchange on a global scale, permeating various aspects of daily life and significantly influencing how individuals perceive and engage with the world around them.

1.3 Impacts of social media platforms

Social media's impact on our daily lives is undeniable and far-reaching. These platforms have integrated into our routines, serving as vital hubs for communication, entertainment, and the exchange of information. From staying connected with others to staying informed about current affairs, social media fulfills myriad roles in our lives. People use these platforms not just for personal interactions but also for professional networking, job searches, and expressing their thoughts and experiences [22].

Social media has undeniably transformed the landscape of business and marketing, extending far beyond mere entertainment. It plays a pivotal role in influencing consumer behavior, enhancing brand visibility, and driving sales. For instance, platforms like Facebook and Twitter have become crucial tools for businesses to engage with customers, build relationships, and gather valuable feedback [52]. Moreover, Social media wields significant influence in cultivating brand communities and nurturing customer loyalty, as evidenced by the impactful campaigns of industry giants such as Starbucks and Nike. These instances showcase the dynamic role of social platforms as vital channels for businesses to engage their desired audience, enhance marketing endeavors, and realize enduring expansion [63].

Beyond individual interactions, social media has catalyzed significant changes on a global scale. Platforms like Twitter and Facebook have been pivotal in amplifying voices during movements like the Gaza humanitarian crisis and the Black Lives Matter protests, showcasing their power to mobilize and unify communities. They have also been instrumental in raising awareness about critical issues such as humanitarian crises, environmental challenges, and political upheavals, transcending geographical barriers to foster global conversations [39].

However, alongside these positive impacts, the rise of social media has introduced complex challenges. Misinformation spreads rapidly, fueled by echo chambers and algorithmic biases. Cyberbullying and privacy breaches have become prevalent concerns, raising questions about digital ethics and responsibility. Moreover, the addictive nature of social media usage has led to concerns about mental health and well-being.

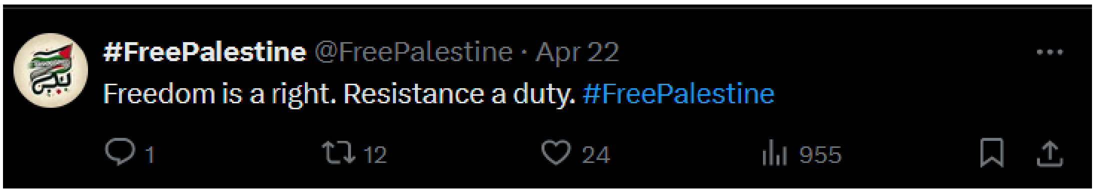
Governments and tech companies are navigating these challenges, striving to strike a balance between regulation and preserving the open nature of these platforms. Efforts to combat misinformation, enhance privacy protections, and promote digital literacy are underway. The goal is to harness the benefits of social media while mitigating its negative repercussions, ensuring that these platforms continue to serve as catalysts for positive change and meaningful connections in our interconnected world [88].

1.4 Twitter

Twitter (Also known as X) is a prominent microblogging platform that has emerged as a powerful force in the realm of social media founded by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams, made its debut in July 2006, offering unique utilities and shaping global events in profound ways [57]. Its core functionality revolves around short messages called *tweets* , Twitter encourages concise yet impactful communication, making it a preferred platform for real-time updates, discussions, and information sharing [49]. According to recent statistics, Twitter boasts over 600 million active users, with approximately 500 million tweets posted daily, making it one of the world's largest social networks as shown in figure 1.1 and the fifth-most visited website globally [78, 85]. This staggering volume of content underscores the platform's significance as a vibrant hub of digital discourse and engagement [47]. Twitter has become a central platform for breaking news, crisis communication, and public discourse on a wide range of topics, including politics, entertainment, sports, and social issues [20, 57]. Its real-time nature enables users to stay informed about unfolding events, engage in discussions, express opinions, and participate in activism and advocacy efforts [62, 71]. Additionally, Twitter plays a vital role in business, marketing, and networking, offering businesses and individuals a platform to connect with audiences, build brand presence, and leverage influencer marketing strategies [11, 58]. Overall, Twitter's utility as a platform for instant communication, information dissemination, and global engagement underscores its transformative influence on society, making it a focal point for studying the dynamics of online discourse, the spread of information, and the intersection of technology and social change [20, 57].

1.4.1 Tweets and Tweet structures

Tweets, the fundamental building blocks of Twitter, encapsulate concise yet impactful messages within the platform's character limit. Originally restricted to 140 characters and later expanded to 280 characters, tweets serve as vehicles for expressing thoughts, sharing information, and engaging with other users. Users can include media attachments such as images,videos,Graphics interchange format (GIF), and polls, enriching the tweet format and allowing for enhanced storytelling and interactive content. Additionally, tweets can be categorized using hashtags (#), making them easily discoverable and contributing to broader conversations.Retweets are a key feature of Twitter, allowing users to share other's tweets on their own timeline, amplifying content and facilitating the viral spread of information within the platform.Likes enable users to express approval or support for tweets, while replies foster conversations and discussions by allowing users to engage directly with each other.Mentions, denoted by (@) followed by a username, create a direct link to the mentioned user's profile within the tweet, encouraging interaction and collaboration among Twitter users.These features, combined with the concise yet expressive nature of tweets, contribute to the dynamic and engaging environment that defines Twitter as a platform for real-time communication and connection.Figure 1.2 displays an example of a tweet posted in April 2024.



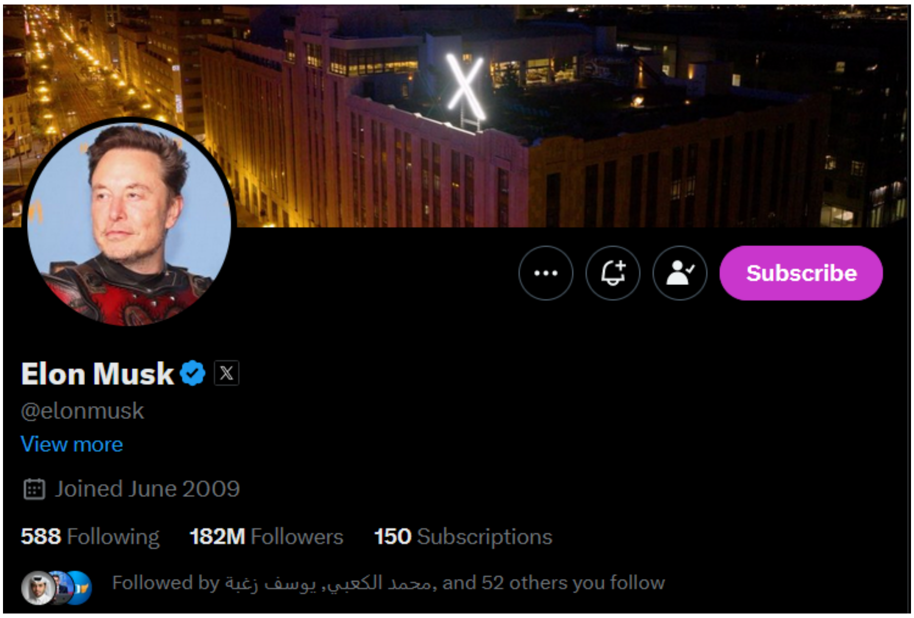
#FreePalestine @FreePalestine · Apr 22

Freedom is a right. Resistance a duty. #FreePalestine

Figure 1.2 – A tweet sample

1.4.2 Twitter profiles

Twitter profiles serve as digital identities that reflect users' personalities, interests, and professional endeavors within the platform's social ecosystem. Each profile is characterized by a unique username preceded by the at symbol (e.g., @username), providing users with a distinct online presence. Profiles typically include a profile picture or avatar, a brief biography that encapsulates users' passions or professional backgrounds, an optional display of their location, and a link to their personal or professional website. The number of followers and accounts followed by a user is prominently displayed, offering insights into their influence and network within the Twitter community. Additionally, Twitter profiles facilitate engagement through features such as tweets, retweets, likes, replies, and mentions, enabling users to connect, share, and interact in a dynamic and collaborative digital environment. Figure 1.3 displays the Twitter profile page for *@elonmusk*, belonging to *Elon Musk*, the current owner of Twitter. The profile page includes the profile picture, header image, and some profile-metadata (i.e., full name, username, number of followers, number of followees, profile creation date, etc.), as of April 2024.



Elon Musk
@elonmusk
View more

Joined June 2009

588 Following 182M Followers 150 Subscriptions

Followed by محمد الكعبي, يوسف زغبة, and 52 others you follow

Figure 1.3 – A Twitter profile sample

1.5 Social Media Bots

The term *Bot* originates from *Software Robot* and encompasses various software systems. It includes automated software agents designed for conversing with humans and compromised accounts used in Command and Control networks to launch attacks [95]. In the context of online social networks (OSN),specific terms like *Social bots* emerge. Some researchers define Social bots as those mimicking human behavior [1], while others specify them as controlling adversary-owned or hijacked accounts on OSNs [18]. However, these definitions may overlook self-declared accounts and the diversity within bot behavior on social media platforms. A refined and commonly accepted definition of social media bots characterizes them as automated accounts with the capability to perform a range of tasks, including posting, commenting, liking, and interacting with other users on social media platforms autonomously, without the need for direct human involvement[70]. This complicates the differentiation between authentic human interactions and automated engagements. In fact, social bots can serve multiple purposes, ranging from facilitating marketing campaigns and disseminating information to amplifying social influence and manipulating public opinion. Consequently, understanding the capabilities, intentions, and implications of these bots has become increasingly important in navigating the modern social media landscape, both benign and malicious, such as spreading information, influencing public opinion, or engaging in coordinated harassment campaigns [38].

The categorization of Social Media Bots (SMB) is essential for identifying and studying them. SMBs can be classified based on their intentions and the degree of imitation of human behavior [82]. Bots can have *benign*, *neutral*, or *malicious* intentions. Examples of benign bots include news bots that automatically post recent articles and chat bots that assist in customer-care services. Malicious SMBs were categorized into different classes. These types often overlap, sharing common objectives and characteristics. Orabi et al. [70] categorize malicious SMBs into four classes: *Spam*, *Social*, *Sybils*, and *Cyborg bots.*

1.5.1 Spam bots

Spam bots are coordinated groups of automated accounts that engage in deceptive and illicit activities on social media platforms or online forums. They operate by spreading misinformation, promoting low-quality or illegal content, or manipulating online platforms for financial gain or other malicious purposes. Spam bots often work in groups, amplifying their reach and impact, and employ tactics such as artificially inflating the visibility of certain content, copying and reposting from legitimate sources while injecting malicious links, or exploiting popular trends or topics to disseminate their content. Their activities are considered fraudulent and can potentially lead users to unwanted or harmful sites or expose them to scams or malware.

1.5.2 Social bots

Social bots are automated accounts or programs designed to operate on social media platforms and online forums with the intention of influencing or manipulating conversations, narratives, and public opinion. They can mimic human behavior and language patterns to appear credible and gain trust within online communities. Social bots can engage in a range of activities, including spreading misinformation and propaganda, engaging in harassment or trolling, distorting public opinion, manipulating narratives, and undermining the authenticity of social media conversations. Their use is often regarded as a threat to the integrity of online discourse, as they can artificially amplify certain messages, infiltrate communities for malicious purposes, shape discussions in an illicit or deceptive manner, and drown out or deflect opposing viewpoints, particularly in the context of political discussions or debates.

1.5.3 Sybils

Sybils are pseudonymous identities or accounts created for malicious purposes on social media platforms, forums, or other online communities. These identities are designed to mislead and deceive by impersonating real users or entities. Sybils can be used for spreading misinformation and propaganda by masquerading as trustworthy sources or influential individuals, manipulating online discussions and artificially inflating the perceived popularity of certain ideas, products, or movements, conducting coordinated attacks such as harassment campaigns or disseminating malware while concealing the true source of the attack, infiltrating online communities and gathering sensitive information by posing as legitimate members, and engaging in fraudulent activities like generating fake reviews, ratings, or endorsements for products or services. Sybils can be challenging to detect because they are designed to appear indistinguishable from genuine accounts, often by cloning or mimicking the profiles and behavior of real users. They can operate individually or as part of larger coordinated networks, amplifying their reach and impact. The use of sybils undermines the integrity and trust of online platforms, as they erode the authenticity of user-generated content and interactions.

1.5.4 Cyborg bots

Cyborg bots in social media refer to automated accounts or profiles that combine both human and machine elements. These bots are typically controlled by humans but use automated processes to engage with users or perform tasks on social media platforms. The term *cyborg* emphasizes the hybrid nature of these bots, as they blend human decision-making and intervention with machine-driven actions. It's important to note that any type of bot can potentially be considered a cyborg if it involves some level of human control or interaction. In other words any class of bots previously mentioned can be cyborg.

1.6 Rising threats of Social Media Bots

Bots are made for specific purposes, and these purposes determine how they behave. Understanding what these bots aim to achieve helps in figuring out what they might do and plan. When bots attack, they often exploit how social networks work. They might take over discussions, send unwanted messages, pretend to be someone else, or mess with how connections are shown [66]. Bots can also pull off what's called a Sybil attack, where they use lots of fake accounts to cause big problems [4]. Bots can do a lot of harmful things like flooding sites with junk content [26], sending out tons of spams, spreading lies to damage trust in information, or trying to trick people into giving away personal information [27, 76, 77].

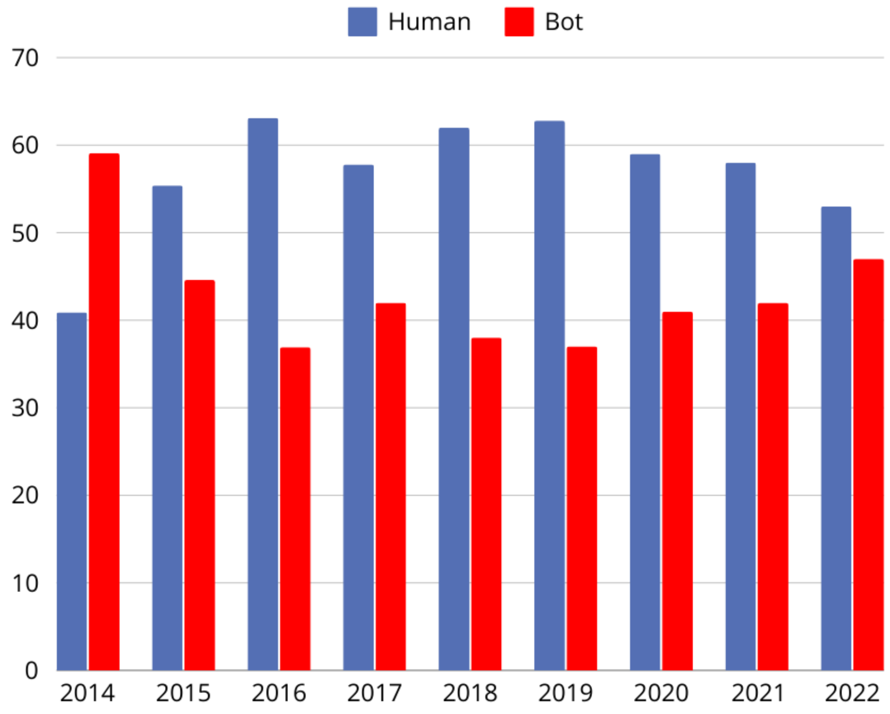


Figure 1.4 – Distribution of bot and human web traffic worldwide over the years [79]

Figure 1.4 displays the distribution of bot and human web traffic worldwide from 2014 to 2022. It shows the percentages of web traffic attributed to humans and bots over this nine-year period. The data reveals that while human traffic has remained consistently higher than bot traffic throughout this timeframe except for 2014, the proportion of bot-generated traffic has been steadily increasing since 2019. Specifically, Twitter managers have stated that 5% of the accounts are fake or spam accounts [83]. However, they acknowledge that the actual number of inauthentic accounts could be higher than this estimate, as detecting them is challenging. The prevalence of bot accounts varies across social media platforms, with studies revealing varying percentages on platforms such as Twitter and Instagram. For instance, benchmark studies indicate that bot accounts on Twitter ranges from 9% to 15% of the total accounts, while approximately 45% of accounts on Instagram are estimated to be bots [83]. These numbers highlight the significant presence of bot accounts on popular social media platforms, impacting user interactions and content dissemination. Many bot creators want to sway what people think, so they often target discussions. Examples of bot attacks detected on social media platforms include:

- During the 2010 US midterm elections, bots on Twitter supported certain candidates and discredited others by linking to websites containing fake news. [73]
- During the 2016 US Presidential elections, about one fifth of the conversation on Twitter was most likely generated by bots. More than 5k Russian bots interfered in the conversation and were removed by Twitter [38].
- In 2017, Social bots produced more than 20 percent of the posts about the Catalan referendum for independence on Twitter [81].
- In 2019, Twitter suspended 5,000 pro-Trump bot accounts that were protesting against the *Russiagate hoax* [23].

1.7 Conclusion

Social media platforms have revolutionized the way we communicate, share information, and engage with the world around us. From facilitating personal connections to enabling global conversations, these digital ecosystems have become integral to our daily lives. The staggering growth and pervasive influence of platforms like Twitter underscore their significance in shaping narratives, fostering trends, and amplifying voices across diverse communities. However, as these platforms have evolved, so too have the challenges they face. The rise of malicious bots poses a substantial threat to the integrity of online discourse and the dissemination of information. These automated entities, with their various intentions and capabilities, have the potential to manipulate public opinion, spread misinformation, and disrupt the very fabric of meaningful dialogue on social media. Recognizing and combating the menace of malicious bots is crucial for preserving the integrity and value of these platforms. As we navigate this digital landscape, it is essential to remain vigilant, promote digital literacy, and advocate for measures that enhance transparency and accountability, ensuring that social media remains a force for positive change and meaningful connections. In the next chapter, we will delve into a comprehensive literature review, exploring the latest research and developments in the field of bot detection on social media platforms. This critical examination will shed light on the techniques, methodologies, and frameworks employed by researchers and practitioners to identify and mitigate the presence of malicious bots, safeguarding the authenticity of online interactions and discourse.

Chapter 2

Social Media Bot Detection

In the realm of social media, the pervasive presence of bots as discussed in Chapter 1 poses a significant threat to the integrity, authenticity, and trustworthiness of online interactions. The proliferation of bots in social media not only distorts the online discourse but also undermines the credibility of information shared within these digital spaces. As the digital landscape continues to evolve, the need to combat the detrimental impact of bots through effective detection mechanisms becomes increasingly imperative. In this chapter, we delve into a comprehensive review focusing on the critical importance of bot detection in social media platforms. By exploring existing research, methodologies, and technologies used in bot detection, we aim to shed light on the evolving strategies and approaches already adopted for identifying and mitigating the influence of bots.

2.1 Importance of Social Media Bot Detection

Social media platforms have become indispensable in today's communication landscape, serving as hubs for sharing information and fostering social connections. However, this open accessibility has also paved the way for the widespread presence of malicious bots, presenting substantial challenges and risks. These malicious bots can pose major security threats by shaping public opinion and spreading false information, disseminating rumors and conspiracy theories, fabricating fake reputations, and undermining political rivals [10]. Consequenctly, maintaining a safe and trustworthy digital environment necessitates robust measures to identify and mitigate the impact of such malicious entities Effective bot detection is crucial for the following reasons:

— *Mitigating the spread of misinformation and disinformation:* Bots have been instrumental in amplifying and disseminating false or misleading information, contributing to the erosion of trust in online content and public discourse. Detecting and mitigating these bots is essential for preserving the integrity of information ecosystems [10, 45].

- *Protecting against manipulation and influence campaigns:* Malicious actors often employ bots to artificially inflate or manipulate public opinion, sway political narratives, and influence decision-making processes. Bot detection can help uncover and counter these coordinated influence campaigns, safeguarding the authenticity of online conversations and debates [29, 45].
- *Combating cyberbullying, harassment, and coordinated attacks:* Bots have been used to launch targeted harassment campaigns, amplify hate speech, and engage in cyberbullying. Detecting and removing these malicious bots can create safer online environments and protect individuals from harmful behavior [10, 37].
- *Ensuring fair and transparent online platforms:* Bots can be used to artificially boost engagement metrics, manipulate trending topics, or distort online discussions. Bot detection is essential for maintaining fair and transparent online platforms, preventing the manipulation of algorithms and ensuring equal opportunities for genuine user participation [29, 45].
- *Preserving trust in social media platforms:* The presence of unchecked bot activity can erode public trust in social media platforms, potentially undermining their credibility and usability. Effective bot detection is crucial for restoring and maintaining user confidence in these platforms [10, 37].
- *Enabling data-driven decision-making:* Social media data is increasingly used for research, marketing, and policy-making purposes. However, the presence of bots can skew data and lead to flawed insights or decisions. Bot detection is necessary to ensure the reliability and validity of data derived from social media sources [10, 37].
- *Protecting user privacy and security:* Some bots may be designed to harvest personal information, distribute malware, or engage in other malicious activities that compromise user privacy and security. Detecting and neutralizing these threats is essential for safeguarding user data and digital safety [29, 45].

2.2 Social Media Bot Detection Methods

Detecting bots on social media platforms has become increasingly important to combat the spread of misinformation, spam, and malicious activities. There are two main approaches to bot detection: *profile-based* and *content-based* [70]. Profile-based approaches analyze the patterns and characteristics of user interactions, such as the frequency of posts, the number of followers, and the network structure. Content-based approaches, on the other hand, focus on the textual or multimedia content shared by users, including the use of specific artifacts such as hashtags, URLs, or sentiment and topic analysis [45]. Machine learning and deep learning techniques have been applied to both approaches, leveraging algorithms like decision trees (DT), support vector machines (SVM), and neural networks

(NN) [10]. For Profile-based detection, these models can learn patterns from historical data on genuine and bot accounts. In content-based detection, natural language processing and computer vision models can identify bot-generated text or images. Figure 2.1 present a hierarchical diagram outlining various techniques used for bots detection.

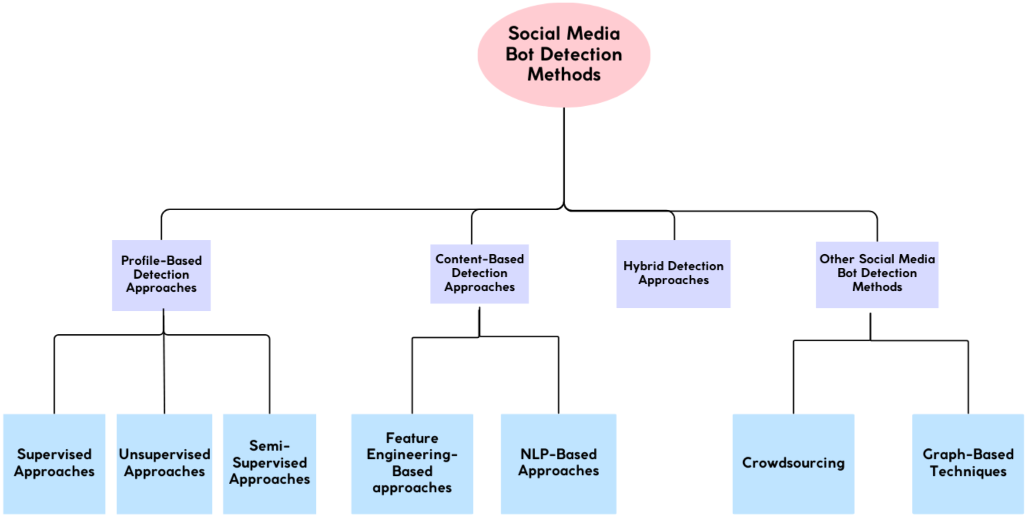

Social Media Bot Detection Methods
Profile-Based Detection Approaches
Content-Based Detection Approaches
Hybrid Detection Approaches
Other Social Media Bot Detection Methods
Supervised Approaches
Unsupervised Approaches
Semi-Supervised Approaches
Feature Engineering-Based approaches
NLP-Based Approaches
Crowdsourcing
Graph-Based Techniques

Figure 2.1 – Classification of existing social media bot detection approaches.

2.2.1 Profile-Based Detection Approaches

Profile-based approaches in bot detection pivot towards scrutinizing the behavioral dynamics exhibited by social media accounts. This scrutiny encompasses analyzing interaction patterns, posting frequency, post timing, engagement levels with other accounts, and assorted behavioral attributes. Through studying these interaction dynamics within the social network, profile-based approaches adeptly unveil anomalies indicative of automated or malevolent behavior [10]. These approaches heavily rely on dissecting profile features and behavioral tendencies of social media accounts to differentiate between human users and bots. Typically, these techniques involve extracting and leveraging various features tied to the account's metadata, activity logs, and social interactions, painting a comprehensive picture for effective bot identification. Some common features used in profile-based approaches include username patterns, account creation date, profile picture, number of posts/tweets [45]. Machine Learning (ML) techniques have been significantly used for bolstering the efficacy of profile-based approaches in detecting bots on social media platforms. By harnessing these technologies, the automatic analysis of interaction patterns, posting frequency, timing of posts, levels of engagement, and other behavioral attributes can be conducted at scale, ensuring more robust bot detection capabilities.

ML-based systems construct predictive models by leveraging historical data, and the accuracy of these models is directly proportional to the volume and quality of data used [75]. ML encompasses a range of techniques including *Supervised*, *Semi-supervised*, and *Unsupervised* learning. All of those techniques have been explored for the behavioral-based SMBs detection. In the following subsections, we briefly outline the principle of each technique, highlighting notable works that have used these methods for detecting SMBs.

2.2.1.1 Supervised Approaches

Supervised machine learning stands as one of the widely explored approaches for behavioral-based bot detection. Researchers have developed numerous supervised machine learning models capable of identifying bot accounts on social media platforms such as Twitter, albeit to varying degrees of success. The key idea is to train a machine learning classifier on a dataset of labeled Twitter accounts, where each account is manually classified, by experts, as either a human or a bot. The classifier learns to recognize patterns and features that distinguish bot accounts from human accounts. Once the classifier is trained, it can then be used to automatically classify new, unlabeled Twitter accounts as either bots or humans. This allows for the rapid and accurate detection of bot accounts, which is crucial for preventing the spread of misinformation, manipulation of public opinion, and other malicious activities by bots on social media [33].

The effectiveness of supervised machine learning techniques for bot detection has been demonstrated in multiple studies such as Cresci et al. [24]. In their study, the authors developed a supervised machine learning based model to detect fake followers on Twitter, which are often generated by bots. Their approach relies primarily on analyzing profile-based features and behavior patterns of followers. By extracting features like *account age*, *default profile image*, *number of friends*, *number of followers*, *follower to friend ratio*, the study aims to enhance bot-follower detection accuracy. Using these profile-based features, they trained different machine learning classifiers like Random Forest (RF), Decision Tree (DT), and Support Vector Machine (SVM) to distinguish between genuine and bot followers. The results indicate that the RF model performed the best, the other models had slightly lower performance, but still achieved respectable performances.

Fonseca Abreu et al. [2] conducted Twitter bot detection experiments using a concise set of 5 features derived from user profile counters, including the *number of tweets*, *followers*, *friends*, *likes*, and *lists the account is included in.* They utilized two spam bot datasets and experimented with various machine learning classifiers, including RF, SVM, Naïve Bayes (NB), and One-Class SVM (ocSVM) models. RF classifier outperformed the other models, demonstrating the highest accuracy, while the ocSVM showed promising results.

Recent studies in bot detection on Twitter have made significant advancements such as

Yang et al. [94]. Their approach utilizes a comprehensive set of 20 user metadata features extracted from Twitter's API, including metrics like the *number of tweets* (including retweets) issued by the user, *number of followers*, *number of friends*, and whether the user *account is verified*. Additionally, they incorporate derived features such as *tweet frequency* and *followers growth rate*, computed from the user metadata. For experimentation, the authors curated a diverse collection of 14 labeled datasets containing both human and bot accounts. They employed a RF classifier, training a separate model on each dataset and testing it on each of the remaining datasets. Furthermore, they applied a feature elimination technique, resulting in the creation of 247 RF models in total. Evaluation of these models was conducted using common performance metrics, allowing for comparison and ranking of model efficacy.

2.2.1.2 Unsupervised Approaches

Unsupervised machine learning is a type of machine learning algorithm that learns patterns from unlabeled data without any prior knowledge or guidance from humans. In unsupervised learning, the algorithms are trained on input data that has not been labeled, classified, or categorized [92]. Leveraging unsupervised techniques for bot detection offers the key advantage of identifying new and evolving bot patterns without the need for extensive manual labeling of training data [93].

Unsupervised algorithms like K-Means [16], Density-based Spatial Clustering of Applications with Noise (DBSCAN) [43], and Hierarchical Clustering (HC) [44] are used to group similar bot accounts or bot-like behaviors into clusters, allowing identification of anomalous or suspicious accounts.

Khalil et al. [54] have used DBSCAN and K-Means to analyze data. They experimented six publicly available datasets with eight profile-based features extracted from the data (e.g., *number of friends* and *followers*, *number of tweets* the user has liked, the *number of public lists* that this user is a member). The researchers found that DBSCAN outperformed K-Means achieving an impressive performance.

Additionally, Principal Component Analysis (PCA) [31] and t-distributed Stochastic Neighbor Embedding (t-SNE) [68], making it easier to identify clusters or outliers. Barhate et al. [14] made a significant contribution by employing an unsupervised machine learning approach. They used PCA and K-Means to categorize users into four groups based on activity-related features (e.g., *length of the string* used to describe the user account, *number of friends* and *followers*, *default profile status*).

2.2.1.3 Semi-supervised Approaches

Semi-supervised learning is a technique that combines both labeled and unlabeled data to improve the learning accuracy of models. It leverages a small amount of labeled data along with a larger amount of unlabeled data to train models, allowing them to generalize better and make more accurate predictions. This approach is particularly useful when obtaining labeled data is expensive or time-consuming, as it maximizes the use of available information to enhance the performance of ML algorithms [101]. In supervised learning, classifiers are trained using labeled data, which can be slow and costly due to the need for manual annotation. Unsupervised learning, on the other hand, works with unlabeled data to identify hidden patterns but may have limited applications and less accurate results. Semi-supervised learning, involves training an initial model on a small set of labeled data and then iteratively applying it to a larger pool of unlabeled data. This method is versatile, applicable to various tasks like classification, regression, clustering, and association, and is particularly useful in scenarios where large amounts of unlabeled data are available, making it cost-effective and practical for real-world applications such as bots detection [87].

A notable work, in this context, is the one presented by Zeng et al. [98] where they applied a semi-supervised self-training approach on a dataset containing both real and fake Twitter accounts. Their suggested technique involved applying a self-training method to automatically categorize Twitter accounts. To mitigate the impact of class imbalance on identification accuracy, they integrated resampling techniques into the self-training process. The proposed framework demonstrated effective identification outcomes across six distinct base classifiers, especially with the initial batch of small-scale labeled Twitter accounts, showcasing promising results.

2.2.2 Content-Based Detection Approaches

Content-based methods for bot detection focus on analyzing the content disseminated via social media accounts, such as tweets. This entails text vectorization using natural language processing (NLP) techniques, as well as feature engineering and extraction from the textual data. These methods entail scrutinizing diverse elements including text, images, links, and other media embedded within shared posts across social media platforms. By discerning various types of shared informationfrom fabricated news and rumors to promotional contentthey can help in distinguishing bots engaged in spreading misinformation, manipulating public opinion, or perpetrating malicious activities [70]. Two primary techniques are commonly used for extracting meaningful data and discerning distinguishing features from social media posts: *feature engineering* and *natural language processing*. The outputs of those techniques are used for building several machine and deep learning models enabling the automatic detection of SMBs.

2.2.2.1 Feature Engineering-based Approaches

In the context of social media bot detection, engineering and extracting features from user posts revolve around identifying specific characteristics that differentiate between human users and bots. These features encompass various aspects of the text data, such as the number of hashtags, links, mentions, and other patterns indicative of automated or human-generated content.

Alarifi et al. [6] conducted a study on detecting *Sybil* accounts, utilizing Twitter4j to collect a dataset of 2000 Twitter accounts comprising humans, bots, and hybrid accounts. They identified eight content-based features such as: retweet percentage, hashtags per tweet, and tweets per day and employed four supervised machine learning algorithms, including RF, Bayesian Network (BN), Sequential Minimal Optimization for Classification (SMO-C), Sequential Minimal Optimization for Regression (SMO-R), and Multi-Layer Perceptron (MLP). Among all the explored classifiers, RF demonstrated the highest performance.

Rajendran et al. [72] achieved a high classification performance using a Bidirectional Long Short-Term Memory (Bi-LSTM) model, effectively distinguishing between bot and genuine accounts based on tweeting rate and frequency. Their study compared various models such as Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Bidirectional Gated Recurrent Unit (Bi-GRU), highlighting the superior performance of Bi-LSTM models in terms of performance.

Alarfaj et al. [5] proposed another methodology for detecting bots on Twitter by analyzing tweet contents collected from both bot and human accounts using the Twitter API. The study focuses on extracting various content-based features, such as *special characters*, *word frequency*, *part-of-speech tags*, and *sentiment analysis*, resulting in 18 features per tweet. To enhance the model's performance, feature selection techniques like Gain Ratio, Information Gain, and Relief-F are used to rank and select the top 10 most relevant features. These selected features are then used as input for multiple machine learning algorithms, including RF, NB, MLP, and Deep Neural Network (DNN), as well as a Rule-based classification algorithm. Among these, the DNN model demonstrated the best overall performance.

2.2.2.2 Natural Language Processing-based Approaches

Natural Language Processing (NLP) is a field of artificial intelligence (AI) and computational linguistics that focuses on enabling computers to understand, interpret, and generate human language. It involves the development of algorithms and models to process and analyze natural language data in various forms, such as text, speech, and more [61]. Text vectorization, on the other hand, is a fundamental technique in Natural Language Processing (NLP) that converts text data into numerical vectors, enabling machine learning algorithms to process and analyze textual information. This approach represents words or documents in a corpus as numerical vectors in a high-dimensional space. Various text vectorization techniques are available, each with unique strengths and weaknesses. These methods transform text into numerical representations, facilitating tasks such as sentiment analysis, text classification, information retrieval, language modeling, and bot detection [3].

— *Bag-of-Words (BoW):* This technique models texts by converting them into fixed-length vectors based on word counts. Each unique word in the corpus is assigned a unique index, creating a vocabulary. For each document, a vector is created with dimensions equal to the vocabulary size. The value at each index in the vector represents the count of the corresponding word in the document. For example, in the context of Twitter, if the vocabulary of all tweet corpus consists of the words ["*happy*", "*birthday*", "*celebrate*"] and the tweet is "*happy happy birthday*", the BoW vector would be [2, 1, 0]. The primary advantage of BoW is its simplicity and ease of implementation. However, it results in high-dimensional, sparse vectors and does not capture the context or semantics of the words [51].

— *Term Frequency-Inverse Document Frequency (TF-IDF):* This method is an extension of the BoW model that considers the importance of words within a document relative to the entire corpus. It aims to reduce the weight of common words and increase the weight of rare but significant words. TF-IDF is calculated as the product of term frequency (TF) and inverse document frequency (IDF). TF measures the frequency of a word in a document, while IDF measures the inverse of the word's frequency across all documents. The formula is given by:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right)$$

Where t is the term, d is the document, N is the total number of documents, and $\text{DF}(t)$ is the number of documents containing term t. The advantage of TF-IDF is that it balances the frequency of terms within documents and across the corpus, providing more informative features. However, it still results in sparse vectors and does not capture the semantic meaning of words [64].

— *Word to Vectors (Word2Vec):* This technique uses a shallow neural network to learn continuous, dense vector representations of words from large corpora of texts, capturing semantic similarities. Word2Vec has two main architectures: Continuous Bag-of-Words (CBOW) and Skip-Gram. CBOW predicts a target word from a set of context words, while Skip-Gram predicts context words from a target word. Words are represented as vectors in a continuous vector space, and words with similar contexts have similar vector representations. For example, the words *tweet* and *post* might have similar vectors due to their contextual usage on Twitter. The primary advantage of Word2Vec is that it captures semantic relationships and context, producing dense, low-dimensional vectors. However, it requires significant computational resources and large amounts of training data [42].

— *Global Vectors for Word Representation (GloVe):* This method combines the advantages of global matrix factorization and local context window methods to generate word vectors. GloVe constructs a word co-occurrence matrix and factorizes it to learn word vectors, while also utilizing local context windows to ensure words that appear together frequently in contexts have similar vectors. The formula is based on the ratio of probabilities of word co-occurrences. For instance, the words *retweet* and *share* might have similar vectors due to their frequent co-occurrence in tweets. GloVe is efficient and scalable, capturing both global statistical information and local context. However, it requires careful tuning and substantial computational resources [15].

Both machine learning and deep learning techniques have undergone extensive exploration within the realm of natural language processing and text vectorization for social media bot detection. Specifically, detecting bots through deep learning and natural language processing has been explored through two main approaches. One approach entails using specialized neural networks designed to analyze patterns and behaviors indicative of bot activity. These networks are trained on extensive datasets to recognize features such as abnormal posting frequencies or repetitive content generation, thereby flagging them as potential bots. Another method involves fine-tuning large language models like Generative Pre-trained Transformer (GPT), Bidirectional Encoder Representations from Transformers (BERT), or other similar models to differentiate between human-generated and bot-generated text based on linguistic cues and contextual understanding.

Cable and Hugh [21] explored various algorithms including NB, SVM, RF, and Long Short-Term Memory (LSTM) for the purpose of detecting political trolls on Twitter. Their study involved comparing the performances of these algorithms. They gathered a dataset comprising tweet IDs associated with the 2016 elections by utilizing the Twitter Application Programming Interface (Twitter API), resulting in a total of 142,560 distinct tweets. Feature extraction was carried out using multiple techniques such as Bag of words, TF-IDF, and Word embeddings. Notably, the SVM model, leveraging character-level TF-IDF features, emerged as the top performer.

Faerber et al. [35] proposed a Convolutional Neural Network (CNN) to detect Twitter bots by analyzing their tweet content. The process starts by cleaning and tokenizing the tweets to create a word dictionary. The text is then vectorized using word embeddings, either trained from scratch on the dataset or utilizing pre-trained Word2Vec embeddings, which serve as the input layer for the CNN model. The model architecture includes two convolutional layers and max-pooling layers, followed by a dense layer and an output layer for classification. The experiments conducted by the authors highlighted the potential of CNNs for detecting Twitter bots based on tweet content.

Another notable contribution by Garcia-Silva et al. [41] involved fine-tuning pre-trained transformer language models such as BERT, GPT, and GPT-2 for bot detection on Twitter. These models begin by tokenizing and converting the input text into numerical vector representations during the pre-processing step. To tailor the models for the specific task, the authors added a classification layer and trained the models on a labeled dataset comprising tweets from both bot and human accounts. After fine-tuning and evaluating the models, they discovered that generative transformer models like GPT and GPT-2 outperformed the bidirectional BERT model in detecting bots, demonstrating the effectiveness of these models in understanding and classifying textual content in this context.

2.2.3 Hybrid Detection Approaches

Hybrid detection approaches integrate both profile-based and content-based methods to enhance the effectiveness of bot detection on platforms like Twitter. By leveraging the strengths of each approach, hybrid approaches can capture a broader spectrum of bot behaviors and characteristics, resulting in more accurate and robust detection. By combining these two approaches, hybrid detection systems can identify bots that may evade detection when only one method is applied. For example, a bot with normal-looking profile metrics but abnormal posting behavior can be detected by the hybrid approach. Similarly, a bot generating sophisticated text content but exhibiting suspicious profile activities can also be flagged.

In this context, Echeverria et al. [32] introduced the *Leave One Botnet Out* (LOBO) framework, which effectively combines profile-based features (such as *number of followers*, *number of friends*, *account age*, and *total number of tweets*) with content-based features extracted from tweets (including the *number of hashtags*, *mentions*, and *URLs*). The authors extracted a comprehensive set of 30 features from user profiles and tweets, then trained multiple tree-based classifiers: DT, RF, XGBoost, LightGBM, and AdaBoost. The results demonstrated the effectiveness of this hybrid approach. Among the experimented classifiers, the LightGBM algorithm performed the best, showcasing the highest accuracy. The other models, while slightly lower in performance, still achieved commendable results, highlighting the robustness and efficiency of combining profile-based and content-based features in detecting botnets. This study underscores the value of hybrid models in enhancing the detection of malicious entities on social media platforms.

Alhassun and Rassam [9] proposed a hybrid framework that combines content-based and profile-based models to detect spam bots on Twitter. The content-based model utilizes a Convolutional Neural Network (CNN) trained on numerical vectors derived from tweets using the Word2Vec technique. The textual content of tweets undergoes several preprocessing steps, including the removal of diacritics and emojis, to enhance the quality of the input data. The profile-based model employs a simple neural network (NN) trained on a variety of features extracted from user profiles and tweets. These features include *account age*, *number of followers*, *number of friends*, and *number of retweets*. By leveraging both content and profile information, the hybrid framework aims to improve detection accuracy. To integrate the two models, the hybrid framework combines the predictions of the CNN and the NN models. The combined outputs are then fed into a fully connected neural network layer with a *softmax* activation function, which classifies the account as either a spam bot or a human. The study's experiments demonstrated that the proposed hybrid framework outperformed both the content-only and profile-based models, showcasing the effectiveness of combining multiple approaches for more accurate bot detection.

Martin-Gutierrez et al. [65] gathered a dataset of 37,438 Twitter accounts using the Twitter API. The study conducted various experiments using different combinations of word-embeddings to create a unified vector based on the textual attributes of the user accounts. These attributes were combined with other profile-based featues like friend count, follower count, username, language, location, and more, to form a potential input vector for a Dense Network called Bot-DenseNet. Comparing these experiments, the Bot-DenseNet achieved the best balance between performance and practicality when using the Robustly Optimized BERT Pretraining Transformer (RoBERTa) as part of the input feature vector.

Alharthi et al. [8] used machine learning to detect malicious Arab Twitter accounts and groups. A dataset of around 4,500 Arab accounts, including 400 identified as part of spam/promotion campaigns, was collected via the Twitter API. The researchers defined a set of 16 features to characterize the accounts, including features related to tweeting behavior (e.g., *tweet rate*, *interval between tweets*), Profile-based features (e.g., *follower-to-friend ratio*), content features (e.g., *presence of URLs* and *hashtags*), and features specifically designed to capture the coordinated behavior of malicious groups (e.g., *retweet rate*, *number of unique accounts retweeted*). Semi-supervised Label Propagation and Label Spreading algorithms were applied, starting with around 500 manually labeled accounts. The models achieved high performance.

2.2.4 Other Social Media Bot Detection Approaches

Bot detection techniques encompass a diverse range beyond just machine learning or deep learning methods. These strategies also integrate *crowdsourcing*, tapping into collective human input to identify patterns and behaviors indicative of bots. Additionally, *graph-based techniques* are used, mapping out connections and interactions among entities to uncover anomalies that might signify bot activity.

2.2.4.1 Crowdsourcing

Crowdsourcing entails a participative online activity where an individual, institution, non-profit organization, or company invites a diverse group of individuals with varying knowledge, skills, and numbers to voluntarily undertake tasks through an open call. These tasks vary in complexity and modularity, and the crowd contributes their work, money, knowledge, and experience, resulting in mutual benefit. Participants receive satisfaction in various forms, such as economic rewards, social recognition, self-esteem, or skill development. Meanwhile, the entity initiating the crowdsourcing effort leverages the contributions of participants to its advantage, tailored to the specific activity undertaken [34]. In the context of social media bot detection, this method requires a substantial investment of either money or time. Nonetheless, certain researchers have adopted this approach to gather labeled datasets for further research purposes.

A notable contribution in this field comes from Wang et al. [89], who employed a social Turing test to evaluate the ability of a trusted online social media user in distinguishing Sybil accounts. Wang et al. [89] proposed a system that utilizes automated algorithms to filter accounts, identifying suspicious ones for further scrutiny by selected crowdworkers whose collective votes enhance the accuracy of Sybil detection. Furthermore, Alarifi et al. [7] engaged ten trained volunteers as crowdworkers to manually classify Twitter accounts as humans, Sybils, or Cyborgs, thereby constructing a reliable ground truth dataset.

2.2.4.2 Graph-Based Techniques

A standard graph structure comprises a set of points referred to as vertices or nodes, positioned within a plane or space, alongside a set of line segments known as edges or links. These edges may connect two nodes or loop back to the same node [91]. These graph structures find extensive applications across various domains for modeling pairwise relationships among entities, including the identification of bots in online social media. This modeling technique is commonly used to depict the structure of social networks. Methods based on graphs leverage the inherent characteristics of social graphs to distinguish bots from genuine users.

Boshmaf et al. [17] introduced a system that employs victim classification to prioritize real users over fake ones, facilitating the detection of fake accounts based on their lower ranking. Initially, supervised machine learning is used to classify victims of social botsthose who interact with fake accounts. The system then utilizes the network structure of online social media, employing a graph-based approach to rank users. This involves assigning lower weights to nodes or users connected to potential victims and conducting short random walks starting from trusted users.

Jinyuan et al. [50] introduced SybilWalk, a novel graph-based approach tailored for detecting Sybil accounts within online social networks. Their methodology involved enhancing the network graph by introducing two distinct labeling nodes: one representing known benign users and the other representing known Sybil accounts. Edges were then established between these new nodes and the user nodes based on a training dataset with labeled instances. Subsequently, random walks were conducted to compute a *badness score* for each node. This score indicates the structural proximity of a given user node to the users directly linked to the labeled Sybil nodes relative to those connected to the labeled benign nodes within the social network. The computed scores serve as a basis for node ranking or classification. One notable advantage of SybilWalk lies in its utilization of labeled benign users and labeled Sybil accounts, allowing it to accommodate weaker homophily and maintain robustness against label noise. The authors assessed the effectiveness of SybilWalk across datasets containing both synthesized and real-world Sybil accounts.

2.3 Conclusion

The ubiquitous presence of bots on social media platforms has emerged as a formidable challenge, threatening the integrity and trustworthiness of online interactions. The study of the literature has demonstrated a multitude of techniques and approaches have been explored to combat the menace of bots, each with its strengths and limitations. From content-based approaches that examine the textual and visual elements shared by accounts, to behavior-based methods that analyze interaction patterns and activity dynamics, researchers have leveraged diverse strategies to unveil the telltale signs of bot behavior. The application of machine learning techniques, including supervised, unsupervised, and semi-supervised methods, has shown promising results in automating the detection process and adapting to evolving bot strategies. Furthermore, the advent of deep learning has ushered in a new era of sophistication, enabling the extraction of intricate features and patterns from multimodal data. Techniques such as recurrent neural networks (e.g., LSTM, GRU, RNN), and large language models (e.g., GPT, BERT, RoBERTa) have demonstrated remarkable efficacy in identifying bots, particularly when combined with ensemble architectures and attention mechanisms. While significant strides have been made, the ever-evolving nature of bots demands continuous innovation and adaptation. Graph-based techniques and crowdsourcing offer alternative avenues for enhancing bot detection capabilities. It's imperative to adopt a holistic and multifaceted approach, leveraging the strengths of various techniques to build robust and effective bot detection systems. In the forthcoming chapter, we will detail the methodology adopted in this thesis for detecting social bots on Twitter. By integrating insights from previous studies, our aim is to make a meaningful contribution to the ongoing fight against the detrimental impact of bots, thereby preserving the authenticity and integrity of online conversations.

Chapter 3

A Hybrid Ensemble Model for Social Media Bot Detection

In the previous chapters, we explored the general landscape of social media platforms and bot activity, and reviewed the most advanced and current techniques in bot detection. Our critical analysis revealed that profile-based detection approaches excel at identifying patterns and anomalies in user account characteristics, such as the age of the account, the frequency of posts, follower/friend ratios, and other metadata. These methods are particularly effective at spotting bots that exhibit unusual profile behaviors, such as newly created accounts or accounts with abnormally high activity levels. However, they often fall short when bots mimic legitimate user profiles or when the profile information is sparse or incomplete. On the other hand, content-based detection approaches focus on the analysis of the actual content posted by users, including the text, links, and multimedia shared. These methods leverage natural language processing and other analytical techniques to detect patterns indicative of bot activity, such as repetitive posting, sentiment analysis, and the use of certain keywords or phrases. While content-based approaches can be highly effective at detecting bots based on the nature of their posts, they can struggle with sophisticated bots that produce content similar to that of human users or when the content is too diverse to establish clear patterns. Consequently, each approach has its unique strengths and limitations, underscoring the need for a hybrid method that combines both profile-based and content-based features to enhance detection accuracy and robustness.

In this chapter, we introduce our proposed methodology, which leverages the complementary potential of combining profile-based and content-based features to enhance the accuracy and robustness of bot detection. This hybrid approach aims to address the deficiencies observed when using either method in isolation, thereby providing a more comprehensive detection mechanism. Hybrid detection approaches integrate profile-based and content-based methods to improve bot detection on platforms like Twitter. It leverage the strengths of both approaches, capturing a wider range of bot behaviors and characteristics for more accurate and robust detection. By merging these methods, hybrid systems can identify bots that might escape detection when only one method is used, providing a refined understanding of bot behavior across both structural and content dimensions. This chapter details the design of our hybrid bot detection system. It outlines the feature extraction process, the machine learning models selection, and the fusion of profile-based and content-based predictions.

3.1 Overview of the Proposed Methodology

The proposed framework for bot detection on Twitter uses a hybrid approach that integrates profile-based features and natural language processing (NLP) techniques for analyzing tweets. Figure 3.1 illustrates the generalized workflow of this methodology.

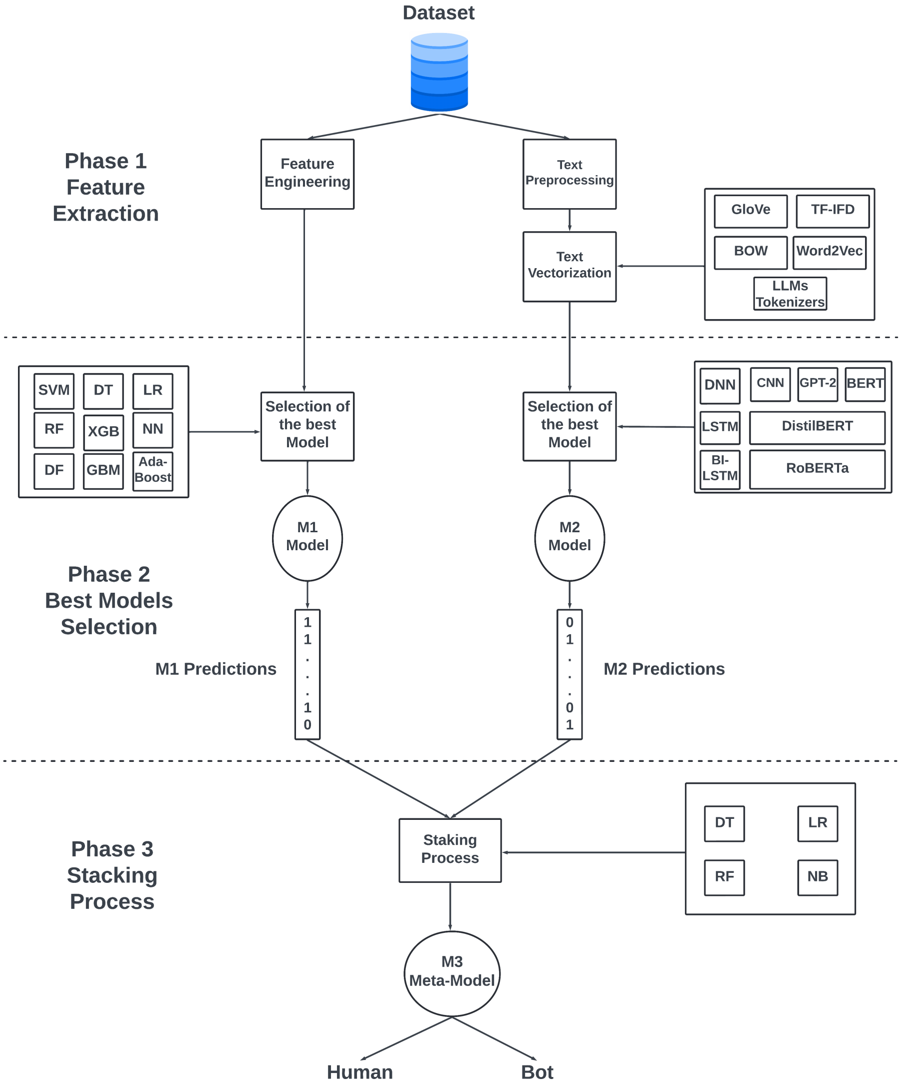


Figure 3.1 – Overview of the adopted methodology.

The proposed framework encompasses two primary machine learning models. The first model, M1, leverages profile-based features extracted from user account information, combined with some statistical features extracted from tweets. These features undergo a selection process to identify the optimal configuration, ensuring that the model effectively differentiates between bot and human accounts based on a comprehensive set of extracted features from both tweets and user profiles. The second model, M2, focuses on the textual content of tweets. This model involves several steps: first, the text data is preprocessed to clean and normalize it, then it is vectorized to convert it into a format suitable for analysis. The processed text is then input into various NLP models. A selection process determines the most effective configuration for M2, ensuring that the chosen model can accurately interpret and classify the tweet content.

After obtaining predictions from the primary models, M1 and M2, these predictions are combined using a stacking technique. This involves feeding the outputs of M1 and M2 into a meta-model, M3. By leveraging the combined insights from both models, the meta-model enhances overall prediction performance, ensuring a more robust detection of bots on Twitter. The final output of this hybrid approach is a bot/human prediction that benefits from the comprehensive analysis of both profile features and tweet content.

3.2 Feature Extraction

Effective feature extraction is essential for building accurate machine learning models. This process entails identifying and transforming pertinent characteristics from raw data into a suitable input representation for the models. In our hybrid framework, we consider both feature engineering and feature vectorization approaches. The M1 model is trained and tested on features extracted from profiles and engineered from the content of tweets, while the M2 model is trained and tested on vectorized tweet contents. In the following sections, we delve into the details of how each of these methods are used in our hybrid framework for bot detection.

3.2.1 Feature Engineering

The feature engineering process involves the transformation of raw data into meaningful features that significantly enhance the performance of machine learning algorithms. In our research, we meticulously identified and engineered a diverse set of features from Twitter data, which serve as indicative signals of bot-like behavior. These features are broadly categorized into two main types: *profile-based* and *content-based*. Profile-based features encompass meaningful attributes extracted from user profiles, such as follower count and account age. These attributes provide insights into the characteristics and activity patterns of Twitter accounts, which can be indicative of bot-like behavior. On the other hand, content-based features are crafted from tweet content itself, capturing various aspects such as the average length of tweets and the frequency of hashtags and

URLs. These features offer insights into the content and engagement patterns of tweets, which can further distinguish between bot and human accounts based on their posting behavior. The success of the M1 model heavily relies on the quality and relevance of these features, as they form the foundation for distinguishing between bot and human accounts. By leveraging both profile-based and content-based features, our model can effectively capture the diverse characteristics and behaviors associated with bot activity on Twitter, enabling more accurate and robust bot detection. In this work, we adopted a set of 25 features (11 profile-based and 14 content-based) that will serve as the foundation for our first machine learning model M1 aimed at detecting bots on Twitter. By combining these complementary feature sets, we aim to capture a comprehensive view of user behavior, enabling our models to leverage the strengths of each approach while mitigating their individual limitations.

3.2.1.1 Selected Profile-based Features

Several profile-based features have gained widespread recognition in the literature for their effectiveness in identifying bots on Twitter, as highlighted in comprehensive review papers [10, 45, 70]. These features provide valuable insights into the behavior, interaction patterns, and account characteristics that are indicative of bot activity. By analyzing such features, we can develop more accurate and reliable models for bot detection, ultimately improving the ability to maintain the integrity and authenticity of interactions on the platform. The widespread adoption of these features is driven by their proven utility in various studies and their contribution to advancing the field of bot detection. Profile-based features can be directly obtained through querying the Twitter API [86], while others can be deduced from available data provided by the Twitter API. Several features from the Twitter API have been eliminated because they either lack significant information or have a weak impact on the bot detection task. Examples include `profile_background_color`, `profile_image_url`, and `profile_sidebar_border_color`, or they have been marked as *deprecated* in the current version of the API, such as `is_translation_enabled` and `is_translator`. By removing these features, the analysis becomes more streamlined and efficient, focusing only on the most relevant data. The following comprises the selected features along with their descriptions and the rationale behind their selection:

— **F1.** `protected`: This binary feature indicates whether the user's Twitter account is protected or not. A value of 1 indicates that the account is protected, which means that the tweets of this account are only visible to approved followers. On the other hand, a value of 0 signifies that the account is public, and its tweets are accessible to anyone on the platform. Bots often opt for public accounts to maximize their reach and dissemination of content to a broader audience. Therefore, the presence of this feature can serve as a crucial indicator in distinguishing between bot and human accounts, as bots typically operate with public visibility to achieve their objectives.

— **F2.** `verified` : Binary features that specifies whether the user's Twitter account is verified or not. Verified accounts are less likely to be bots, as the verification process typically involves manual checks.

— **F3.** `followers_count` : A numerical feature that represents the number of followers a user has on their Twitter account. Bots often exhibit abnormal patterns in the number of followers they have, either having an extremely high or extremely low number of followers compared to regular human users.

— **F4.** `friends_count` : Provides to the number of other users that a given user is following on Twitter. Similar to the `followers_count` feature, bots may exhibit unusual patterns in the number of accounts they follow. Bots may follow a large number of accounts indiscriminately or follow very few accounts, deviating from typical human behavior.

— **F5.** `friends_to_followers_ratio`: This numerical feature represents the ratio of the number of accounts a user is following (`friends_count`) to the number of followers the user has (`followers_count`). It is calculated by dividing `friends_count` by `followers_count`. The `friends_to_followers_ratio` can provide valuable insights into the interaction patterns and behavior of a user on Twitter, which can be useful in distinguishing between genuine human accounts and potential bot accounts. The mathematical formula for calculating the `friends_to_followers_ratio` is:

$$friends\_to\_followers\_ratio = \frac{friends\_count}{followers\_count}$$

— **F6.** `listed_count` : This numerical feature represents the number of public Twitter lists that the user is a member of. Being included in lists is often an indication of the user's influence or relevance within a particular topic or community. Bots are less likely to be included in many lists, as their content and interactions may be perceived as less valuable or authentic by human users. A low `listed_count` value can be a potential indicator of bot behavior.

— **F7.** `account_age` : Represents the age of a user's Twitter account, calculated in days. It is derived from the `created_at` feature provided by the Twitter API, which is a timestamp representing when the user's account was created. The `account_age` feature provides insights into the longevity of a user's presence on Twitter, which can be a useful indicator in distinguishing between genuine human accounts and potential bot accounts. The `account_age` is calculated as follows:

$$account\_age = current\_date - created\_at$$

Where *current_date* is the current date for which the calculation is being performed (in the same date format as *created_at*).

- **F8.** `favourites_count` : A numerical feature that showcase the number of tweets that the user has marked as favorites. Favoring tweets is a way for users to engage with and appreciate content on Twitter. Bots may exhibit unusual patterns in favoring tweets, either favoring an excessive number of tweets in a short period of time or rarely favoring any tweets at all. Abnormal `favourites_count` values can potentially signify bot-like behavior.
- **F9.** `statuses_count_per_day` : This feature represents the average number of tweets a user posts per day. It is derived from the `statuses_count` feature, which represents the total number of tweets posted by the user, and the `account_age` feature, which represents the age of the user's account in days. The rational behind adopting this feature is based on the fact that human users typically exhibit varied posting patterns, with intermittent activity and fluctuating post frequencies over time. In contrast, bots often maintain a consistent and elevated posting rate, particularly those engaged in spamming or automated propaganda dissemination. By adopting this feature, anomalies such as unnaturally high or rigidly consistent posting frequencies can be identified, suggesting bot activity. The `statuses_count_per_day` feature is calculated as follows:

$$statuses\_count\_per\_day = \frac{statuses\_count}{account\_age}$$

  Where *statuses_count* is the total number of tweets that the user has posted from their account provided by the Twitter API. *account_age* is the age of the user's account in days (see **F7**).
- **F10.** `default_profile` : This binary feature indicates whether the user is using the default Twitter profile settings. The `default_profile` settings include the standard color scheme, layout, and other default options provided by Twitter. Human users tend to personalize their profiles by changing these settings to reflect their preferences and individuality. Bots, being automated accounts, may not prioritize customizing their profiles beyond the default settings. Therefore, a user with a default profile can be a potential signal of bot-like behavior.
- **F11.** `default_profile_image`: This feature denotes whether the user is using the default Twitter profile picture (typically an egg or a generic silhouette image) or has uploaded a custom profile picture. Human users often upload personalized profile pictures that represent their identity or interests. Bots, being automated accounts, may not prioritize changing the default profile picture, as their primary purpose is often not to establish a personal online presence. The presence of a default profile picture can be an indicator of bot activity.

3.2.1.2 Proposed Content-based Features

In addition to the selected profile-based features from the previous section, we have identified and extracted several content-based features that can potentially enhance the performance of bot detection models. These proposed features aim to capture unique patterns and characteristics that may be indicative of bot-like behavior on Twitter. By incorporating these features into our models, we aim to improve the accuracy and robustness of bot detection, leveraging insights that have not been explored in previous researches. The proposed features are derived from a careful analysis of Twitter data and an in-depth understanding of bot behavior, enabling to complement the existing feature set and potentially uncover new signals that can aid in distinguishing bots from human users more effectively. Below are the proposed features, along with their descriptions and the reasoning for their selection:

— **F12.** `avg_nb_mentions` : This feature represents the average number of mentions (@username) per tweet posted by the user. It provides insights into the user's interaction patterns and engagement with other users on Twitter. A high `avg_nb_mentions` value may indicate a user who frequently mentions or engages with others, potentially suggesting human-like behavior. On the other hand, a low `avg_nb_mentions` value could be indicative of bot-like behavior, where the account primarily broadcasts content without engaging with other users. This feature is calculated as follows :

$$avg\_nb\_mentions = \frac{\sum_{i=1}^{N} mentions\_in\_tweet_i}{N}$$

Where $mentions\_in\_tweet_i$ is number of mentions in each tweet i, and N is total number of tweets for a target user.

— **F13.** `avg_nb_hashtags` : Represents the average number of hashtags (#hashtag) per tweet posted by the user. Hashtags are commonly used to categorize tweets and participate in ongoing conversations or trends. A high `avg_nb_hashtags` value may suggest a user who actively participates in discussions and follows trends, potentially indicating human-like behavior. Conversely, a low `avg_nb_hashtags` value could be associated with bot-like behavior, where the account primarily posts content without engaging in broader conversations. The `avg_nb_hashtags` is calculated as follows :

$$avg\_nb\_hashtags = \frac{\sum_{i=1}^{N} hashtags\_in\_tweet_i}{N}$$

Where `hashtags_in_tweet`$_i$ is number of hashtags in each tweet i, and N is total number of tweets for the user..

— **F14.** `avg_tweet_length` : This feature represents the average length of tweets posted by the user, typically measured in characters. It provides insights into the user's tweeting style and the complexity of the content they share. A high `avg_tweet_length` value may indicate a user who posts more detailed and verbose content, potentially suggesting human-like behavior. On the other hand, a low `avg_tweet_length` value could be associated with bot-like behavior, where the account posts shorter, more concise, or automated content. The calculation of `avg_tweet_length` is given by the formula:

$$avg\_tweet\_length = \frac{\sum_{i=1}^{N} tweet\_length_i}{N}$$

Where $\texttt{tweet\_length}_i$ is length of each tweet, and N is total number of tweets.

- **F15.** `avg_nb_words` : This feature corresponds to the average number of words per tweet posted by the user. Similar to `avg_tweet_length`, it provides insights into the user's tweeting style and the complexity of the content they share. A high `avg_words` value may indicate a user who posts more detailed and verbose content, potentially suggesting human-like behavior. Conversely, a low `avg_words` value could be associated with bot-like behavior, where the account posts shorter, more concise, or automated content.The formula for determining `avg_words` is:

$$avg\_nb\_words = \frac{\sum_{i=1}^{N} words\_in\_tweet_i}{N}$$

  Where $\texttt{words\_in\_tweet}_i$ is number of words in each tweet i, and N is total number of tweets for the user.

- **F16.** `avg_nb_elongated_words` : This numerical feature represents the average number of elongated words (words with repeated letters) per tweet posted by the user. Elongated words are often used to emphasize or exaggerate specific words, which is a common trait in human-like communication. A high `avg_elongated_words` value may indicate a user who employs more expressive language, potentially suggesting human-like behavior. On the other hand, a low `avg_elongated_words` value could be associated with bot-like behavior, where the account posts more formal or automated content.`avg_elongated_words` is derived using the equation:

$$avg\_nb\_elongated\_words = \frac{\sum_{i=1}^{N} elongated\_words\_in\_tweet_i}{N}$$

  Where $elongated\_words\_in\_tweet_i$ is number of words elongated in each tweet i, and N is total number of tweets of the user.

- **F17.** `avg_nb_exclamation_marks` : This numerical feature represents the average number of exclamation marks (!) per tweet posted by the user. Exclamation marks are often used to convey emphasis or strong emotions, which is a common trait in human-like communication. A high `avg_exclamation_marks` value may indicate a user who employs more expressive language, potentially suggesting human-like behavior. Conversely, a low `avg_exclamation_marks` value could be associated with bot-like behavior, where the account posts more formal or automated content. The formula for determining `avg_exclamation_marks` is:

$$avg\_nb\_exclamation\_marks = \frac{\sum_{i=1}^{N} exclamation\_marks\_in\_tweet_i}{N}$$

Where $exclamation\_marks\_in\_tweet_i$ is number of exclamation marks in each tweet i, and N is total number of tweets for the user.

— **F18.** `avg_nb_question_marks` : This feature numerical represents the average number of question marks (?) per tweet posted by the user. Question marks are often used to ask questions or express uncertainty, which is a common trait in human-like communication and engagement. A high `avg_question_marks` value may indicate a user who employs more interactive language and engages in conversations, potentially suggesting human-like behavior. Conversely, a low `avg_question_marks` value could be associated with bot-like behavior, where the account primarily broadcasts content without engaging in discussions or seeking input. The calculation of `avg_question_marks` is given by the formula:

$$avg\_nb\_question\_marks = \frac{\sum_{i=1}^{N} question\_marks\_in\_tweet_i}{N}$$

— **F19.** `avg_nb_dots`: This numerical feature represents the average number of dots (.) per tweet posted by the user. Dots are often used in abbreviations, ellipses, or to separate parts of a message, which can be a common trait in both human-like and bot-like communication. The interpretation of this feature may depend on the specific context and other accompanying features. The `avg_dots` can be determined by the following equation:

$$avg\_nb\_dots = \frac{\sum_{i=1}^{N} dots\_in\_tweet_i}{N}$$

Where $dots\_in\_tweet_i$ is number of dots in each tweet i, and N is total number of tweets.

— **F20.** `avg_nb_capitalized_words`: This feature represents the average number of capitalized words per tweet posted by the user. Capitalized words are often used to emphasize or highlight specific words or phrases, which is a common trait in human-like communication. A high `avg_capitalized_words` value may indicate a user who employs more expressive language, potentially suggesting human-like behavior. Conversely, a low `avg_capitalized_words` value could be associated with bot-like behavior, where the account posts more formal or automated content. The formula for determining `avg_capitalized_words` is:

$$avg\_nb\_capitalized\_words = \frac{\sum_{i=1}^{N} capitalized\_words\_in\_tweet_i}{N}$$

Where $capitalized\_words\_in\_tweet_i$ is number of capitalized words in each tweet i, and N is total number of tweets.

— **F21.** `avg_nb_positive_emoticons`: This feature represents the average number of positive emoticons (e.g., :), :-), ;), ;-)) per tweet posted by the user. Emoticons are often used to convey emotions, sentiments, or reactions, which is a common trait in human-like communication. A high `avg_positive_emoticons` value may indicate a user who employs more expressive language and emotive content, potentially suggesting human-like behavior. Conversely, a low `avg_positive_emoticons` value could be associated with bot-like behavior, where the account posts more formal or automated content. The calculation of `avg_positive_emoticons` is given by the formula:

$$avg\_nb\_positive\_emoticons = \frac{\sum_{i=1}^{N} positive\_emoticons\_in\_tweet_i}{N}$$

Where $positive\_emoticons\_in\_tweet_i$ is number of positive emoticons in each tweet i, and N is total number of tweets of the user.

— **F22.** `avg_negative_emoticons`: This feature represents the average number of negative emoticons (e.g., :(, :-(, ;(, ;-() per tweet posted by the user. Similar to `avg_positive_emoticons`, it provides insights into the user's use of expressive language and emotive content. A high `avg_negative_emoticons` value may indicate a user who employs more expressive language and emotive content, potentially suggesting human-like behavior. Conversely, a low `avg_negative_emoticons` value could be associated with bot-like behavior, where the account posts more formal or automated content. The `avg_negative_emoticons` can be determined by the following equation:

$$avg\_nb\_negative\_emoticons = \frac{\sum_{i=1}^{N} negative\_emoticons\_in\_tweet_i}{N}$$

Where $negative\_emoticons\_in\_tweet_i$ is of number of negative emoticons in each tweet i, and N is total number of tweets of the user.

— **F23.** `avg_nb_emojis`: This feature represents the average number of emojis per tweet posted by the user. Emojis are widely used in modern communication to convey emotions, sentiments, or reactions, which is a common trait in human-like communication. A high `avg_emojis` value may indicate a user who employs more expressive language and emotive content, potentially suggesting human-like behavior.

Conversely, a low `avg_emojis` value could be associated with bot-like behavior, where the account posts more formal or automated content. The `avg_emojis` is derived using the equation:

$$avg\_nb\_emojis = \frac{\sum_{i=1}^{N} emojis\_in\_tweet_i}{N}$$

Where $emojis\_in\_tweet_i$ is of number of emojis in each tweet i, and N is total number of tweets of the user.

— **F24.** `avg_nb_urls` : This feature represents the average number of URLs per tweet posted by the user. URLs are often shared to provide additional information, resources, or links to external content. A high `avg_nb_urls` value may indicate a user who shares a significant amount of external content or resources, which could be associated with both human-like and bot-like behavior, depending on the specific context and purpose of the account. Conversely, a low `avg_nb_urls` value could suggest an account that primarily posts original content or engages in discussions without sharing external links. To calculate `avg_nb_urls`, we used the following formula:

$$avg\_urls = \frac{\sum_{i=1}^{N} URLs\_in\_tweet_i}{N}$$

Where $URLs\_in\_tweet_i$ is of number of URLs in each tweet i, and N is total number of tweets of the user.

— **F25.** `retweet_ratio` : This feature represents the ratio of retweets among the total number of posted tweets by the user. Retweeting is a way for users to share and amplify content posted by others on Twitter. The `retweet_ratio` feature provides insights into the user's engagement patterns and the extent to which their content is being shared and propagated within the Twitter community. It can be determined by the following equation:

$$retweet\_ratio = \frac{\sum_{i=1}^{N} retweet_i}{N}$$

Where $retweet_i$ is the status of a tweet i indicating whether the tweet is originated from the user or simply retweeted, and N is total number of tweets of the user.

3.2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) plays a crucial role in enhancing the effectiveness of our methodology, particularly in optimizing the performance of the M2 model. It enables the identification of nuanced patterns and relationships within tweets, aiding the model in distinguishing between tweets generated by humans and those produced by bots. This phase consists of two pivotal steps: *Text Preprocessing* and *Text Vectorization.* Through the application of these NLP techniques, our goal is to maximize the performance of the intended M2 model, allowing it to efficiently learn patterns and features from tweet data. Ultimately, this empowers the model to accurately detect bot activity on Twitter.

3.2.2.1 Text Preprocessing

Text preprocessing plays a crucial role in natural language processing which make this step a vital component of our methodology and necessary for accurately detecting and classifying bots based on their content. As mentioned in Chapter 1, tweets have a unique structure and can contain various textual elements such as mentions, retweets, hashtags, and links. Since every Twitter user can write freely, tweets are often informal and unstructured. By employing preprocessing techniques, we cleanse and standardize textual content to focus on the meaningful aspects of tweets. This involves breaking down the text into meaningful units, reducing noise forms, and normalizing the vocabulary. Such rigorous preprocessing ensures that the data used for bot detection algorithms is consistent, informative, and ready for advanced analysis techniques, ultimately aiding in the precise identification and mitigation of bot activity on Twitter. In our methodology we used the following steps :

1. *Lowercasing*: Convert all characters to lowercase to maintain uniformity. This ensures that the text is processed consistently, regardless of case variations, which is crucial for accurate bot detection.

   Example: Before After

   "Hello World!" "hello world!"

2. *URL Handling*: Remove all URLs from the tweet and replace them with the special token "`url`". URLs can be used for malicious purposes, and uniform handling helps in identifying patterns typical of bot activity.

   Example: Before After

   "Check this out: http://example.com" "Check this out: `url`"

3. *Retweet Handling*: Remove all retweet signs (RT) from the tweet and replace them with the special token "`retweet`". Retweets can indicate bot behavior aimed at amplifying specific messages.

   Example: Before After

   "RT @user: Check this out!" "`retweet` @user: Check this out!"

4. *Mentions Handling*: Remove all mention signs (@) from the tweet and replace them with the special token "`mention`" to anonymize tweets. Bots often use mentions to amplify their reach, so handling mentions uniformly aids in detecting such behavior.

Example: Before After

"Thanks @user for the info!" "Thanks `mention` for the info!"

5. *Numbers Handling*: Remove all numbers from the tweet and replace them with the special token "`number`". Numbers can vary widely and may obfuscate patterns; standardizing them helps in recognizing bot-like numeric usage.

Example: Before After

"I have 2 cats." "I have `number` cats."

6. *Punctuation Handling*: Remove all punctuation marks from the tweet except the following, which will be replaced by special tokens corresponding to each punctuation:

   — "`question`" for (?)

   — "`exclamation`" for (!)

   — "`dot`" for (.)

   For multiple punctuation marks which refer to the repetition of a single type of punctuation mark used consecutively (more than once) in a tweet, replace them with special tokens as follows:

   — "`MultiQuestionMarks`" for (???)

   — "`MultiExclamationMarks`" for (!!!)

   — "`MultiStopMarks`" for (...)

   This kind of exaggerated punctuation used for emphasis and often used by human.

Example: Before After

"What??? Really!!! I can't believe it..." "What `MultiQuestionMarks` Really `MultiExclamationMarks` I can't believe it `MultiStopMarks`"

7. *Emojis Handling*: Remove emoticons and non-ASCII characters except the most popular emojis, Example of some of the emojis handled:

   — "`smile`" for 🙂

   — "`heart`" for ❤

   — "`laughing`" for 😀

   — "`angryface`" for 😠

   Emojis can be significant in user interactions; handling them consistently helps in distinguishing between human and bot behavior.

Example: Before After

"I'm so happy 🙂❤" "I'm so happy `smile heartemo`"

8. *Contraction Replacement*: Expand contractions to their full forms. Bots often use formal language, so normalizing contractions can help in identifying less human-like text.

Example: Before After

"doesn't" "does not"

9. *Slang/Abbreviation Replacement*: Replace slang and abbreviations with their full forms. Bots might not use slang correctly, so expanding these can help in distinguishing between human and bot text.

Example: Before After

— 'btw' 'by the way'

— 'lol' 'laughing out loud'

— 'omg' 'oh my god'

10. *Tokenization*: Split the tweet into individual tokens (words). Tokenization is essential for text analysis, allowing for detailed examination of each word or token, which is crucial for identifying bot-like patterns.

Example: Before After

"This is an example." ["This", "is", "an", "example", "."]

11. *Stop Word Removal, Stemming, and Lemmatization*: Remove stop words, and apply stemming and lemmatization to the tokens. This reduces the text to its core components, making it easier to detect unnatural or repetitive patterns typical of bots.

Example: Before After

"This is an example" "exampl"

3.2.2.2 Text Vectorization

Once the text has been preprocessed, the next step involves preparing the preprocessed tweets for classification using various text vectorization techniques. As discussed in Chapter 2, these methods are essential for transforming raw text into numerical vector representations, which can be efficiently processed by the models. Our approach encompasses several vectorization approaches, including Term Frequency-Inverse Document Frequency (TF-IDF) [64], Bag of Words (BoW) [51], Word to Vectors (Word2Vec) [42], and Global Vectors for Word Representation (GloVe) [15], as detailed in Section 2.2.2.2.

These techniques play a vital role in converting textual data into structured numerical inputs that enable effective machine learning analysis and classification.

On the other hand, Large Language Models (LLMs) are fundamentally different in how they process and represent text data. Unlike traditional and deep models that often rely on vectorization techniques to vectorize text data. LLMs have their own built-in tokenization mechanisms. These tokenizers break down the input text into smaller units called *tokens*, which can be individual words, subwords, or even characters, depending on the specific model architecture. This allows LLMs to capture complex patterns and relationships within the tweets without the need for explicit text vectorization techniques [30].

In our methodology, while traditional text vectorization techniques are crucial for preprocessing and feature extraction with other models, we also explore the capability of fine-tuning LLMs directly on tokenized text. Fine-tuning LLMs enables us to leverage the vast pretraining knowledge of these models and adapt them to the specific task of tweet classification. By fine-tuning LLMs on tokenized tweet data, we aim to harness their ability to capture intricate linguistic patterns and nuances inherent in social media text, ultimately improving the performance of our classification models for identifying bot activity on Twitter.

3.3 Selection of Best Models

As described in Section 3.1, the proposed methodology involves training two distinct models, previously referred to as M1 and M2, using a diverse range of approaches. An extensive exploration of various machine learning classifiers and deep learning models was conducted. For M1, the focus was on evaluating both traditional classifiers and more advanced models. For M2, the emphasis was on effectively modeling the textual content of tweets using techniques specific to natural language processing.

3.3.1 Explored Classifiers

In the pursuit of optimizing M1's classification capabilities, we conducted an extensive evaluation of various classifiers. The exploration included several traditional machine learning classifiers: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Multilayer Perceptron (MLP), eXtreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM), and Deep Forest (DF) [99, 100]. While the initial classifiers are well-known and widely used, Deep Forest stands out due to its unique approach. DF is an ensemble learning algorithm that integrates the principles of traditional decision tree ensembles with deep learning techniques. Developed by Zhou and Feng [99, 100], this method aims to provide a powerful alternative to deep neural networks, especially in scenarios with limited data or computational resources. DF operates through a multi-grained scanning process that captures valuable features from raw input data using sliding windows of various sizes. This generates diverse feature representations, which are then processed through a cascade of decision tree ensembles. Each layer of the cascade refines the input features by incorporating the outputs of the previous layer, progressively enhancing the model's understanding of complex patterns. The use of multiple decision tree models at each layer enhances robustness and generalization, while an adaptive layer growth mechanism ensures that the model's depth is optimized based on performance improvements, preventing unnecessary complexity. Overall, the selection of these classifiers was based on their proven effectiveness in handling various classification tasks. Our goal was to identify the most suitable classifier for the specific requirements of M1's classification needs. After selecting the classifiers, we trained them using the previously engineered features extracted from the dataset.

Similar to the selection of M1, we conducted a thorough exploration of different models for M2. This exhaustive search involved evaluating various deep learning models to determine the most effective approach for modeling the textual content of tweets. The process begins with comprehensive text preprocessing and vectorization techniques, as detailed in the preceding section. Subsequently, we investigated a range of models to identify the most suitable one: For M2, the emphasis is on effectively modeling the textual content of tweets. This process begins with comprehensive text preprocessing and vectorization techniques, as detailed in the preceding section. Subsequently, a range of models is investigated to determine the most suitable model as following :

- *Convolutional Neural Networks (CNNs)*: Known for their ability to capture spatial features effectively, CNNs are well-suited for analyzing text data with a structured layout.
- *Deep Neural Networks (DNNs)*: Offer the advantage of learning complex hierarchical representations of the text, enabling them to discern subtle nuances and patterns within tweets.
- *Long Short-Term Memory Networks (LSTM):* Adept at retaining important information over extended sequences, LSTMs are beneficial for understanding the context and sentiment evolution in tweets.
- *Bidirectional Long Short-Term Memory Networks (Bi-LSTM)*: Combine the strengths of LSTMs with bidirectional processing, allowing them to capture both past and future context, enhancing their understanding of tweet content.

In addition to these models, we employed a novel approach in natural language processing through the use of Large Language Models (LLMs). LLMs, such as BERT, RoBERTa, DistilBERT, and GPT-2, are pre-trained models that utilize deep learning techniques to process, generate, and analyze text based on vast amounts of training data. These models employ transformer-based neural network architectures with billions of parameters to capture complex patterns and relationships in textual data [60]. A key aspect of these models is the attention mechanism, which includes the use of an attention mask. The attention mask is a binary tensor that indicates which tokens in the input sequence should be attended to and which should be ignored, typically used to differentiate between actual content and padding tokens. This ensures that padding does not affect the models performance, allowing it to focus only on relevant information [36]. BERT, RoBERTa, DistilBERT, and GPT-2 were specifically chosen due to their accessibility as free resources and their proven effectiveness in handling complex language structures. To mitigate computational challenges, we utilized smaller versions of these models, ensuring robust performance while maintaining efficiency for our tweet classification objectives. This comprehensive evaluation allowed us to leverage the strengths of each model type, ultimately enhancing the accuracy and robustness of our hybrid framework in detecting bot-like behavior on Twitter.

3.3.2 Hyperparameter tuning

Throughout the training process, we performed a comprehensive hyperparameter tuning strategy to optimize the performance of candidate models for M1. Each classifier had its unique set of parameters requiring careful adjustment to achieve peak performance. To this end, we utilized both Randomized Search and Grid Search methodologies. Randomized Search [12] was particularly favored for its efficiency in exploring a wide range of hyperparameter combinations within limited computational resources. This method randomly samples from a predefined range of hyperparameters, providing a broad exploration and quickly identifying promising regions in the hyperparameter space. On the other hand, Grid Search [97] systematically examines all possible combinations within a specified grid of hyperparameters. While more exhaustive and computationally intensive, Grid Search ensures that no potential optimal configuration is overlooked. By strategically adopting these methodologies, we systematically explored various hyperparameter settings to identify the most effective configurations for each classifier. This meticulous approach ensured that the candidate models were finely tuned to meet the specific requirements of their classification tasks.

Hyperparameter tuning is also a critical process in optimizing the performance of candidate deep learning models for M2. Hyperparameters are configuration settings that govern the learning process and architecture of these models. Key hyperparameters include the learning rate, which determines the step size during training; the batch size, which specifies the number of samples propagated through the network in each iteration; and the number of layers and their configurations, such as filter sizes, pooling sizes, and dropout rates. Additionally, hyperparameters encompass the choice of activation functions, which introduce non-linearities into the model, enabling it to learn complex patterns in the data. Properly tuning these hyperparameters is essential for achieving optimal model performance and preventing issues like overfitting or underfitting[96]. To select the best M2 model, we employed a comprehensive approach that included not only general hyperparameter tuning but also fine-tuning of large language models (LLMs). Fine-tuning involves modifying the final layer(s) of a pre-trained model for a specific task, such as tweet classification. This process allows the model to adapt its extensive pre-trained knowledge to the target domain or objective, leveraging the robust language understanding and generation abilities acquired during its initial training on vast datasets. Fine-tuning significantly enhances the model's performance on specialized tasks by tailoring it to the nuances and requirements of the specific application, thereby ensuring both accuracy and efficiency. This dual approach of hyperparameter tuning and fine-tuning LLMs ensures that M2 is optimized to its full potential, delivering superior performance in tweet content classification [30].

3.3.3 Validation Process

To reliably evaluate the performance of our models and mitigate the risk of overfitting, we employed cross-validation, a widely used technique in machine learning [48]. Cross-validation provides an unbiased estimate of each model's generalization capabilities on unseen data by partitioning the dataset into multiple subsets, known as folds. During each iteration of the cross-validation process, one fold is held out as the validation set, while the remaining folds are used for training. This procedure is repeated multiple times, with each fold serving as the validation set exactly once. By averaging the performance metrics across all iterations, cross-validation provides a robust assessment of the model's performance across different subsets of the data. This helps to ensure that the evaluation results are not overly influenced by the particular choice of training and validation data, thereby providing a more accurate reflection of the model's true capabilities. However, it's worth noting that cross-validation wasn't utilized during the fine-tuning of the large language models (LLMs) due to computational resource constraints. Fine-tuning LLMs requires significant computational resources and time, especially when considering the vast amount of data and the complexity of the models involved. Instead, a train test split was performed to identify the best-performing model for each task.

Each experiment was carefully monitored, and the performance of the models was assessed using appropriate evaluation metrics. The goal was to select the model configuration that achieved the best balance of performance and efficiency for the specific task at hand. Specifically, we computed various performance metrics, on held-out test sets, allowing for a comprehensive assessment and informed model selection. The metrics used for evaluation are:

- **Recall** (also known as Sensitivity or True Positive Rate): Recall is the fraction of correctly classified bot accounts out of the total actual bot accounts. It is calculated as:

$$Recall = \frac{correctly\ predicted\ bots}{total\ number\ of\ bots}$$

- **Precision**: Precision is the fraction of correctly classified bot accounts out of the total accounts classified as bots. It is calculated as:

$$Precision = \frac{correctly\ predicted\ bots}{total\ number\ of\ accounts\ predicted\ bots}$$

- **Accuracy**: Accuracy is the fraction of correctly classified Twitter accounts (as bot or human) out of the total number of accounts. It is calculated as:

$$Accuracy = \frac{correctly\ predicted\ bots + correctly\ predicted\ humans}{total\ number\ of\ accounts}$$

- **F1-score**: The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both measures. The F1-score is often used for evaluating the performance of models dealing with imbalanced datasets (e.g., when the number of bot accounts is significantly different from the number of human accounts). It is calculated as:

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Through comparing the performance of all the explored classifiers and models, the classifier or model that demonstrates the highest performance across multiple evaluation metrics is selected as the best model for M1 and M2, respectively. This careful evaluation process ensures that the chosen models are not only accurate but also robust in their ability to detect bots effectively on Twitter. By combining the strengths of different classifiers and models, further enhances the overall performance and reliability of bot detection systems in the dynamic social media landscape.

3.4 Stacking Process

After training and selecting the best configurations for M1 and M2, we employed a stacking technique to combine their predictions and further enhance the overall performance of our hybrid bot detection system. Stacking is an ensemble learning technique that amalgamates the predictions of multiple models to create a more robust and accurate predictive model. The core idea behind stacking is to leverage the strengths of different models and mitigate their individual weaknesses by integrating their outputs through a meta-model [19]. Here's how the stacking process works in our methodology:

1. We independently train the selected best models M1 and M2 on the same dataset, utilizing their respective feature sets.
2. The predictions generated by M1 and M2 for each instance in the dataset are recorded and treated as new feature vectors. These predictions are then combined to form a new dataset that includes the original features along with the predictions from M1 and M2.
3. The new dataset formed by combining the predictions from M1 and M2, is then utilized to train a meta-model, denoted as M3. Similar to the selection process for M1 and M2, M3 is chosen after training various machine learning classifiers and evaluating their performance across key metrics (see section 3.3.3).
4. During the training process of M3, it learns to combine the predictions from M1 and M2 optimally, effectively acting as an ensemble that capitalizes on the strengths of both models while compensating for their individual weaknesses.
5. Once trained, M3 can be utilized to make final predictions by incorporating the original features along with the predictions from M1 and M2 as input. This enables us to leverage the collective knowledge of all three models in making informed decisions.

3.5 Conclusion

In this chapter, we presented our proposed methodology for detecting bots on Twitter using a hybrid ensemble approach. By combining profile-based features and natural language processing techniques for analyzing tweet content, our framework aims to leverage the strengths of both methods while mitigating their individual limitations. The methodology comprises several key components: feature engineering to extract relevant profile-based and content-based features, natural language processing techniques for text preprocessing and vectorization, machine learning model selection and hyperparameter tuning for the primary models M1 and M2, and a stacking ensemble process to combine the predictions from M1 and M2 using a meta-model M3. Through this hybrid approach, we seek to capture a comprehensive representation of user behavior on Twitter, encompassing both structural patterns in user profiles and nuanced patterns in the language and content of tweets. By fusing these complementary perspectives, our methodology strives to enhance the accuracy and robustness of bot detection, enabling more effective identification of sophisticated bots that may evade detection by solely relying on either profile-based or content-based methods. In the upcoming chapter, we will delve into the results and experiments stemming from our proposed methodology for bot detection on Twitter. This will involve a detailed analysis of the performance metrics obtained through our hybrid ensemble approach, comparing them against existing methods and benchmarks.

Chapter 4

Experimentation and Results

In the previous chapter, we outlined our proposed hybrid ensemble methodology for detecting bots on Twitter. This approach aims to leverage the complementary strengths of profile-based features and natural language processing techniques by combining their respective predictions through a stacking ensemble process. The effectiveness of any machine learning model or methodology ultimately lies in its ability to generalize well to unseen data and deliver robust performance in real-world scenarios. Consequently, rigorous experimental evaluation is imperative to assess the validity and practical utility of our proposed bot detection framework. In this chapter, we delve into the experimental setup and evaluation process undertaken to validate our methodology. We begin by introducing the dataset employed for training and testing our models, highlighting its unique characteristics and suitability for the task of bot detection on Twitter. The data preparation process, including any preprocessing steps and partitioning techniques, is described in detail to ensure transparency and reproducibility. Subsequently, we present the results obtained from our experiments, examining the performance of the individual models M1 and M2, as well as the stacked ensemble model (meta-model) M3. A comprehensive analysis of these results is provided, shedding light on the strengths and limitations of each approach, and elucidating the potential benefits of our hybrid ensemble strategy. To establish a baseline for comparison, we contrast the performance of our methodology against existing state-of-the-art techniques for bot detection on Twitter. This comparative analysis aims to highlight the advantages and potential improvements offered by our proposed approach, contributing to the advancement of this rapidly evolving field. Additionally, we evaluated the models on diverse datasets to ensure the generalizability of the framework. Finally, we conclude the chapter by summarizing the key findings and insights gained from our experimental evaluation, setting the stage for a broader discussion on the implications and future directions of this research in the subsequent chapters.

4.1 Data Preparation

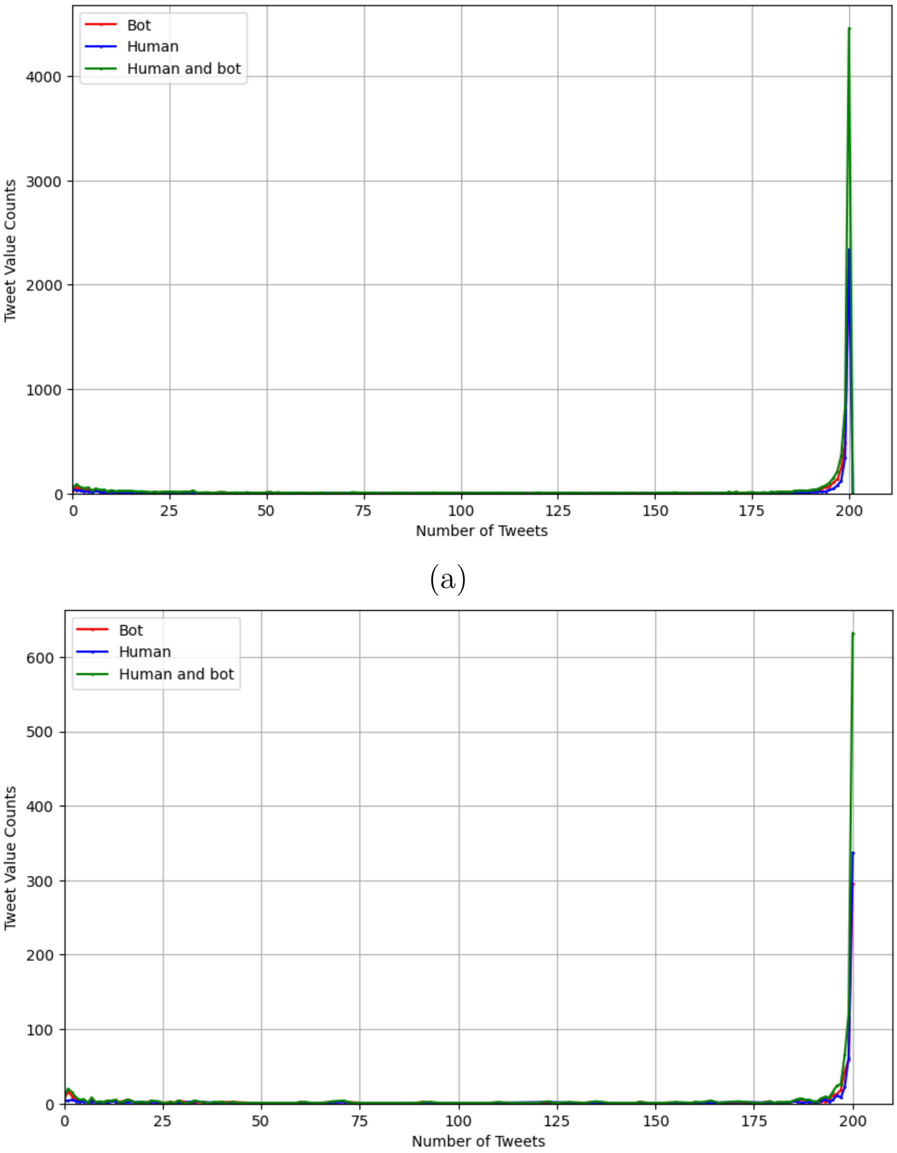
After exploring various datasets, we found that TwiBot-20 is the most suitable for hybrid approaches. TwiBot-20 is specifically designed to overcome the limitations of previous Twitter bot detection datasets, which suffered from low user diversity, limited user information, data scarcity, and questionable reliability. Older datasets are especially unsuitable for hybrid approaches because they were typically designed for either profile-based methods or content-based methods, but not both. Additionally, these older datasets contain outdated data, which further limits their effectiveness. As bots have evolved significantly, relying on outdated datasets has a negative impact on the ability to detect modern bots, making TwiBot-20's up-to-date and comprehensive data crucial for current bot detection methods. The TwiBot-20 dataset is distinguished by several key characteristics:

- *User Diversity*: The users in TwiBot-20 are diverse in terms of geographic locations and interest domains, ensuring better representation of the real-world Twittersphere. The users were collected through breadth-first search starting from diverse seed users across politics, business, entertainment, and sports domains.
- *Diverse Set of Features*: TwiBot-20 provides a diverse set of features spanning user activities and profile characteristics, allowing for the development and evaluation of bot detection methods that can effectively capture the multifaceted nature of Twitter bots.
- *Multi-modal User Information*: TwiBot-20 includes multiple modalities of user information: Content-based (tweets), Profile-based. This comprehensive user information enables leveraging various approaches, including those based on user activities and profile features.
- *Trustworthy Annotations*: The dataset was carefully annotated through a specialized strategy involving crowdsourcing, manual verification, and cross-checking with known bot characteristics. The annotations are generally trustworthy and consistent with previous literature on bot behavior.
- *Imbalanced Distribution*: Table 4.1 showcases the distribution of users in Twibot-20 datsaet in both the train and test sets. As shown in the table, there is an imbalance between Bot class and human class. The test set size of 1182 users is approximately 14.3% of the train set size of 8277 users, which is within the commonly recommended range of 10-20% for splitting datasets. Notably, the ratio of human to bot users is similar in both sets, with a ratio of approximately 1.28:1 in the train set and 0.85:1 in the test set, maintaining a consistent imbalance ratio.
- *Rich Dataset*: Figures 4.1 shows that both the train and test datasets are rich in terms of number tweets with most of users have around 200 tweet, with around

Dataset	**User Type**	**Number of Users**	**Total**
Twibot-20-Train	Human	4646	**8277**
	Bot	3632	
Twibot-20-Test	Human	543	**1182**
	Bot	640	

Table 4.1 – Distribution of Users in Twibot-20

1,398,410 tweets in the train set and 198,597 tweets in the test set. Additionally, the distribution of tweets is similar in both sets.



(a)

(b)

Figure 4.1 – Distribution of tweets in the Twibot-20 dataset: (a) train set, (b) test set

By utilizing the TwiBot-20 dataset, this thesis aims to evaluate and develop robust bot detection methods that can handle the diversity and complexity of real-world Twitter bots. Those key characteristics of TwiBot-20 make it a valuable resource for studying and benchmarking Twitter bot detection approaches. To test the generalizability of our framework, we also evaluate its performance on two additional datasets: Cresci2017 [28] and Cresci2015 [25]. The Cresci2017 dataset is a large-scale collection of Twitter accounts, including genuine accounts, social spambots, traditional spambots, and fake followers. It contains over 13,000 accounts and over 18 millions manually annotated tweets, providing a diverse range of bot types for evaluation. The Cresci2015 dataset, on the other hand, focuses on fake followers and includes 5301 accounts and more that 2.8 million tweets. These datasets offer different challenges and bot types, allowing us to assess our model's performance across various scenarios and bot evolution stages. It is important to note that both Cresci2017 and Cresci2015 datasets have limitations when it comes to hybrid approaches. The primary issue is that the account information is not directly linked to the corresponding tweets. This disconnection between user profiles and their content makes it challenging to combine feature-based and content-based analyses effectively. As a result, these datasets are not ideal for evaluating hybrid approaches that aim to leverage both profile characteristics and tweet content simultaneously. Due to these limitations, both Cresci2017 and Cresci2015 datasets are specifically used to evaluate the performance of our M2 model, which focuses solely on content-based analysis. Table 4.2 and table 4.3 provides a statistical view of both datasets.

	Accounts	**Tweets**
genuine accounts	3,474	8,377,522
social spambots #1	991	1,610,176
social spambots #2	3,457	428,542
social spambots #3	464	1,418,626
traditional spambots #1	1,000	145,094
traditional spambots #2	100	74,957
traditional spambots #3	433	5,794,931
traditional spambots #4	1,128	133,311
fake followers	3,351	196,027
Total	**14,398**	**18,179,186**

Table 4.2 – Description of the Cresci2017 Dataset

4.2 Feature Engineering based Detection

In this section, we offer a comprehensive analysis of the engineered features, highlighting their impact in distinguishing between bots and human accounts. Additionally, we present

Sub-Datasets	**Accounts**	**Tweets**
TFP	469	563,693
E13	1,481	2,068,037
FSF	1,169	22,910
INT	1,337	58,925
TWT	845	114,192
Total	**5,301**	**2,827,757**

Table 4.3 – Description of the Cresci2015 Dataset

and discuss the results obtained when these features are used as inputs to various machine learning classifiers, providing insights into their effectiveness and performance.

4.2.1 Feature Analysis

Feature analysis refers to the process of understanding and selecting the most relevant features from a dataset for building predictive models. It involves various techniques to identify the attributes that contribute the most to the prediction task while eliminating irrelevant features. The importance of feature analysis cannot be overstated, as it directly impacts the performance, accuracy, and interpretability of the models. By focusing on meaningful features, we can enhance model efficiency, reduce overfitting, and gain insights into the underlying patterns and relationships within the data, ultimately leading to more robust and reliable predictions.

Figure 4.2 displays the information gain of various features used for tweet analysis or user profiling. Information Gain (IG) [67] is a measure used in machine learning to quantify the effectiveness of a feature in splitting a dataset into classes. The features are ranked based on their relative importance or predictive power, with the bars extending further to the right indicating higher information gain and thus more valuable features. The `verified` status of an account stands out as the most influential feature, surpassing all other profile-based and content-based features in information gain. Notably, several content-based features are among the top ten in terms of information gain. These include `avg_nb_capitalized_words`, `avg_nb_dots`, `avg_nb_mentions`, `avg_nb_emojis`, and `avg_nb_question_mark`. While profile-based features such as `followers_count`, `friends_to_followers_ratio`, `account_age`, and `favourites_count` are highly informative, content-based features like `avg_nb_capitalized_words`, `avg_nb_dots`, `avg_nb_mentions`, `avg_nb_emojis`, `avg_nb_urls`, and `avg_nb_question_marks` also show substantial predictive power, indicating a balanced importance of both types of features in the model. While the `protected` feature has an information gain of 0, indicating it does not contribute to the predictive model's performance, it is still important to consider it for data Sensitivity and Privacy reasons, `protected` indicate that their tweets

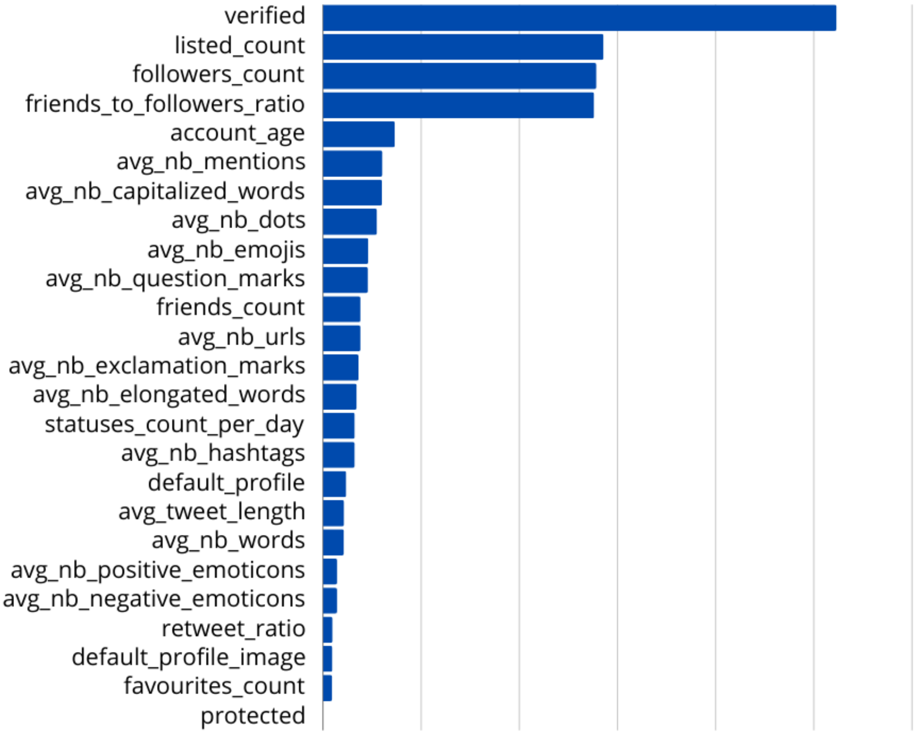


Figure 4.2 – Information Gain of Twitter User and Content Features.

are private and only visible to approved followers. This can affect the availability and nature of the data that can be collected from these accounts. Understanding whether an account is protected is crucial for respecting user privacy and adhering to data protection regulations.

4.2.2 Results and Analysis

In this section, we present and discuss the performance of various machine learning classifiers experimented with for determining the M1 model of our framework across several evaluation metrics. In this analysis, the F1-score is considered the primary metric for comparing model performance due to the imbalanced nature of the Twibot-20 dataset.

Table 4.4 illustrates the performance metrics of various classifiers trained on engineered features across two scenarios: using solely profile-based features and incorporating both profile-based and content-based features. In the first scenario, LR, SVM and MLP showcase consistent performance, boasting a higher recall of 81.66% and higher precision of 86.30%, AdaBoost has the second higher recall and precision, very close to the top performing classifiers. DT provides the lowest score with 72.39% of F1-score. In the second scenario, DT and SVM achieve a quasi-flawless recall of 99.38% and 99.53% respectively post the integration of content-based features, albeit with slightly lower precision and F1-score compared to other models. But SVM precision have been considerably decreased. However, LR, SVM and MLP scores have changed upon the inclusion of content-based features.

Model	Profile-based features				Profile + Content-based features			
	R (%)	P (%)	A (%)	F1 (%)	R (%)	P (%)	A (%)	F1 (%)
DT	72.36	72.47	72.36	72.39	99.38	75.27	81.99	85.66
LR	81.66	**86.30**	81.66	80.70	95.31	78.01	82.92	85.79
SVM	81.66	**86.30**	81.66	80.70	**99.53**	75.65	82.42	85.96
AdaBoost	81.49	84.83	81.49	80.71	95.00	78.55	83.26	86.00
MLP	81.66	**86.30**	81.66	80.70	93.59	79.87	83.77	86.19
RF	80.47	82.20	80.47	79.94	93.59	**80.19**	84.02	86.37
DF	80.47	82.20	79.02	79.95	93.91	80.03	84.02	86.41
XGB	78.95	79.77	78.95	78.57	96.72	78.16	83.60	86.45
GBM	81.07	84.05	81.07	80.32	95.63	79.69	**84.45**	**86.93**

Table 4.4 – Performance of models trained on engineered features: R: Recall, P: Precision, A: Accuracy, F1: F1-score

SVM stands out with a remarkable recall of 99.53%, followed by DT at 99.38% and XGB at 96.72%. Notably, SVM model experiences a considerable decrease in precision. DF yields a high F1-score of 86.41% with a slight decrease in precision. AdaBoost displays a noteworthy increase in recall, surging to 95% while witnessing a marginal decline in precision from 84.83% to 78.55%. GBM emerges as the top performer, achieving the highest F1-score of 86.93% post the incorporation of content-based features, coupled with a commendable precision of 79.45% and an accuracy of 84.45%. These results underscore GBM's effectiveness in striking a balance between identifying bot accounts and minimizing false positives. Hence, GBM stands out as the optimal model, demonstrating a robust balance among precision, recall, and F1-score by harnessing both profile and content-based features. Therefore, it is deemed the prime candidate for M1 within our framework.

Table 4.5 summarizes the parameter values used in *RandomizedSearch* for the machine learning classifiers explored in this study. Each classifier has specific hyperparameters that can be tuned to enhance its performance. The table provides a comprehensive list of these hyperparameters along with their respective value ranges or sets of possible values. Additionally, Table 4.6 details the parameters of the best-performing pipeline for the GBM classifier.

Model	**Parameter**	**Values**
SVM	`C`	[0.1, 1, 10, 100, 1000]
	`gamma`	[1, 0.1, 0.01, 0.001, 0.0001]
	`kernel`	['rbf']
XGBoost	`learning_rate`	[0.01, 0.05, 0.1, 0.2]
	`max_depth`	[3, 4, 5, 6, 7, 8]
	`subsample`	[0.6, 0.7, 0.8, 0.9, 1.0]
	`colsample_bytree`	[0.3, 0.4, 0.5, 0.6, 0.7]
	`n_estimators`	[100, 200, 300, 400, 500]

	objective	['binary:logistic']
	gamma	[0, 0.1, 0.2, 0.3, 0.4]
	reg_alpha	[0, 0.1, 0.5, 1.0]
Random Forest	n_estimators	[50, 100, 200, 300, 400, 500]
	max_features	['auto', 'sqrt', 'log2']
	max_depth	[None, 10, 20, 30, 40, 50]
	min_samples_split	[2, 5, 10]
	min_samples_leaf	[1, 2, 4]
	bootstrap	[True, False]
AdaBoost	n_estimators	[50, 100, 200, 300, 400, 500]
	learning_rate	[0.01, 0.05, 0.1, 0.2, 0.5, 1.0]
MLP	hidden_layer_sizes	[(64,), (128,), (256,)]
	activation	['relu', 'tanh', 'logistic']
	solver	['adam', 'sgd']
	alpha	[0.0001, 0.001, 0.01]
	learning_rate	['constant', 'adaptive']
	max_iter	[200, 500, 1000]
Logistic Regression	C	np.logspace(-4, 4, 20)
	penalty	['l1', 'l2']
Decision Tree	max_depth	[3, 5, 7, 9, None]
	min_samples_split	[2, 5, 10, 20]
	min_samples_leaf	[1, 2, 4, 8]
	max_features	['sqrt', 'log2', None]
	criterion	['gini', 'entropy']
GBM	learning_rate	[0.01, 0.05, 0.1, 0.2]
	max_depth	[3, 4, 5, 6, 7, 8]
	max_features	[0.3, 0.5, 0.7, 0.9, 1.0]
	min_samples_leaf	[1, 3, 5, 7, 9]
	min_samples_split	[2, 4, 6, 8, 10, 12]
	n_estimators	[50, 100, 150, 200, 250, 300]
	subsample	[0.2, 0.4, 0.6, 0.8, 1.0]
Deep Forest	n_estimators	[50, 100, 200]
	max_layers	[10, 20, 30]
	n_trees	[1, 2, 3, 4]
	n_trees_in_layer	[100, 200, 300]
	min_samples_leaf	[1, 2, 4, 6, 8]
	max_features	[None, 'sqrt', 'log2']

Table 4.5 – Grid parameters for various models used in RandomizedSearchCV

Parameter	**Value**
learning_rate	0.01
max_depth	8
max_features	0.7
min_samples_leaf	7
min_samples_split	12
n_estimators	100
subsample	0.2

Table 4.6 – Parameters of the Best Pipeline for GradientBoostingClassifier

4.3 NLP-based Detection

Tables 4.7–4.8 show the performance of different deep learning models (CNN, LSTM, Bi-LSTM, and DNN) using TF-IDF and BoW representation, respectively, with different N-gram ranges (1-1, 1-2, 2-2, 1-3, 2-3, 3-3, 1-4, 4-4) and we temporarily set the maximum number of features (tokens) extracted by BOW and TF-IDF to 5000 based on experimental values.

For the TF-IDF representation, the DNN model outperformed all other models in most cases except when using 3-grams. With 1-4-grams, the DNN model demonstrated the highest performance, achieving a recall of 78.91%, precision of 84.87%, and an F1-score of 81.78%. The DNN model consistently outperformed other models across all evaluation metrics. The Bi-LSTM model also performed well, achieving the second-highest F1-scores overall and outperforming the DNN model when using 3-grams.

N-grams	Model	Recall (%)	Precision (%)	Accuracy (%)	F1-score (%)
1-1	CNN	65.40	67.20	66.80	66.00
	LSTM	73.46	73.42	73.46	73.35
	Bi-LSTM	74.39	74.35	74.39	74.36
	DNN	**77.81**	**75.67**	**74.97**	**76.73**
1-2	CNN	68.40	70.20	69.80	69.00
	LSTM	77.03	75.96	74.39	76.49
	Bi-LSTM	78.75	81.42	78.78	80.06
	DNN	**81.71**	**81.71**	**80.37**	**81.71**
2-2	CNN	68.40	70.20	69.80	69.00
	LSTM	77.03	75.96	74.39	76.49
	Bi-LSTM	78.75	81.42	78.78	80.06
	DNN	**82.81**	**80.67**	**79.97**	**81.73**

2-3	CNN	68.40	70.20	69.80	69.00
	LSTM	74.22	76.37	73.63	75.28
	Bi-LSTM	79.06	77.37	76.16	78.21
	DNN	**79.35**	**79.16**	**77.60**	**79.35**
1-3	CNN	68.40	70.20	69.80	69.00
	LSTM	70.90	75.40	73.20	73.00
	Bi-LSTM	73.50	77.80	75.90	76.20
	DNN	**82.81**	**80.67**	**79.97**	**81.73**
3-3	CNN	64.40	66.77	66.12	65.50
	LSTM	69.69	73.11	69.74	71.36
	Bi-LSTM	**77.34**	**80.49**	**77.60**	**78.88**
	DNN	72.19	76.49	72.95	74.28
1-4	CNN	64.40	66.77	66.12	65.50
	LSTM	74.38	79.20	75.57	76.71
	Bi-LSTM	77.34	80.49	77.60	78.88
	DNN	**78.91**	**84.87**	**80.98**	**81.78**
4-4	CNN	64.40	66.77	66.12	65.50
	LSTM	75.31	71.09	70.08	73.14
	Bi-LSTM	74.22	71.75	70.25	72.96
	DNN	**75.31**	**74.27**	**72.53**	**74.79**

Table 4.7 – Performance of DL models using TF-IDF

For the Bag of Words (BoW) representation with 1-3 grams, the DNN model achieved the highest recall at 81.90%, precision at 81.78%, and F1-score at 81.90%. Both the DNN and Bi-LSTM models stood out with high scores across various evaluation metrics, underscoring their effectiveness in handling the 1-3 gram representation. The DNN model consistently outperformed all other models, highlighting its superiority in managing the 1-3 gram range. Overall, the 1-3 gram range yielded the best performance across different models and evaluation metrics, indicating its robustness for this task.

N-grams	Model	Recall (%)	Precision (%)	Accuracy (%)	F1-score (%)
1-1	CNN	67.40	69.20	68.80	68.00
	LSTM	73.75	77.00	73.88	75.34
	Bi-LSTM	75.78	**79.77**	76.50	77.72
	DNN	**82.50**	79.64	**79.12**	**81.04**
1-2	CNN	67.40	69.20	68.80	68.00

	LSTM	80.63	78.90	77.85	79.75
	Bi-LSTM	**83.44**	78.53	78.70	80.91
	DNN	79.69	**84.16**	**80.90**	**81.86**
2-2	CNN	67.40	69.20	68.80	68.00
	LSTM	74.06	75.12	72.70	74.59
	Bi-LSTM	73.44	76.42	73.37	74.90
	DNN	**78.12**	**80.39**	**77.85**	**79.24**
1-3	CNN	70.40	71.77	71.12	70.50
	LSTM	78.44	81.76	78.87	80.06
	Bi-LSTM	81.72	79.48	78.70	80.59
	DNN	**82.03**	**81.78**	**80.39**	**81.90**
2-3	CNN	70.40	71.77	71.12	70.50
	LSTM	80.63	79.02	77.94	79.81
	Bi-LSTM	**81.72**	**79.48**	**78.70**	**80.59**
	DNN	80.16	79.04	77.77	79.60
3-3	CNN	70.40	71.77	71.12	70.50
	LSTM	73.44	75.44	72.70	74.43
	Bi-LSTM	**77.66**	74.07	73.20	**75.82**
	DNN	72.50	**77.59**	**73.80**	74.96
1-4	CNN	66.40	68.77	68.12	67.50
	LSTM	83.59	79.85	79.71	81.68
	Bi-LSTM	73.44	82.02	76.92	77.79
	DNN	**81.56**	**81.75**	**80.30**	**81.75**
4-4	CNN	70.40	71.77	71.12	70.50
	LSTM	68.75	74.83	70.58	71.66
	Bi-LSTM	70.31	**76.01**	**71.94**	**73.05**
	DNN	**70.47**	74.79	71.17	72.57

Table 4.8 – Performance of DL models using BoW

When exploring Word2Vec and GloVe as vectorization techniques, see Table 4.9, the results indicate that the DNN model surpasses the performance of other models with Word2Vec achieving an F1-score of 75.13%, while GloVe combined with a CNN reached an F1-score of 70.88%. However, it's noteworthy that neither vectorization technique outperformed the results obtained using the BoW approach.

Table 4.10 summarizes the performance of the four fine-tuned language models used to determine the M2 model: GPT-2, BERT, RoBERTa, and DistillBERT. Fine-tuning

	Model	Recall (%)	Precision (%)	Accuracy (%)	F1-score (%)
Word2Vec	CNN	62.66	70.85	65.85	66.50
	LSTM	86.25	61.68	63.57	71.92
	Bi-LSTM	86.09	63.04	65.17	72.79
	DNN	**76.25**	**74.05**	**72.70**	**75.13**
GloVe	CNN	**72.66**	**69.20**	**67.71**	**70.88**
	LSTM	67.97	64.93	62.81	66.41
	Bi-LSTM	72.97	68.27	67.03	70.54
	DNN	66.41	64.01	61.62	65.18

Table 4.9 – Performance of DL models using Word2Vec and GloVe

involves taking a pre-trained language model and further training it on our dataset. This process allows the model to adjust its parameters to better fit the nuances and specific patterns of the target domain. The results indicate that all the explored language models performed below expectations across the evaluation metrics, underscoring the challenges in applying these models to our specific task of bot detection. Despite the fine-tuning process, which typically enhances model performance by adapting it to domain-specific data, the models struggled to achieve satisfactory results, highlighting the complexity of the bot detection problem in our context.

Model	Recall (%)	Precision (%)	Accuracy (%)	F1-score (%)
GPT-2	60.90	72.70	69.80	66.30
BERT	65.80	**74.68**	71.20	68.90
RoBERTa	73.40	67.20	**73.50**	70.10
DistillBERT	**81.23**	70.00	72.52	**75.20**

Table 4.10 – Performance of fine-tuned LLMs

Among the models, DistillBERT achieved the highest recall at 81.23%, demonstrating its effectiveness in correctly identifying a higher proportion of bot accounts compared to the other LLMs. BERT, however, achieved the highest precision at 74.68%, indicating it had the highest proportion of accurate bot account predictions, thus suggesting fewer false positives. DistillBERT also recorded the highest F1-score of 75.20%, indicating the best balance between recall and precision among the evaluated models. Nevertheless, the overall performance was lower than anticipated, with none of the models achieving an F1-score above 76%. Two primary factors contribute to this subpar performance. First, the dataset might not be sufficiently large to effectively fine-tune a large language model. Fine-tuning LLMs typically requires extensive datasets to capture the nuanced patterns within the data, and our dataset might lack the necessary volume for robust training. Second, none of the fine-tuned models were pre-trained specifically on tweet data. Tweets often contain unique language patterns, abbreviations, hashtags, and informalities that general-purpose language models might struggle to handle without further domain-specific training.

Upon comparison of the outcomes, the DNN model utilizing a 1-3 grams representation through BoW exhibited superior performance. To further optimize its efficacy, we experimented the DNN model with various sets of max features values (256, 512, 1024, 2048, 4096, 5000, 10000) to determine the optimal value. The results of these experiments are illustrated in the Figure 4.3.

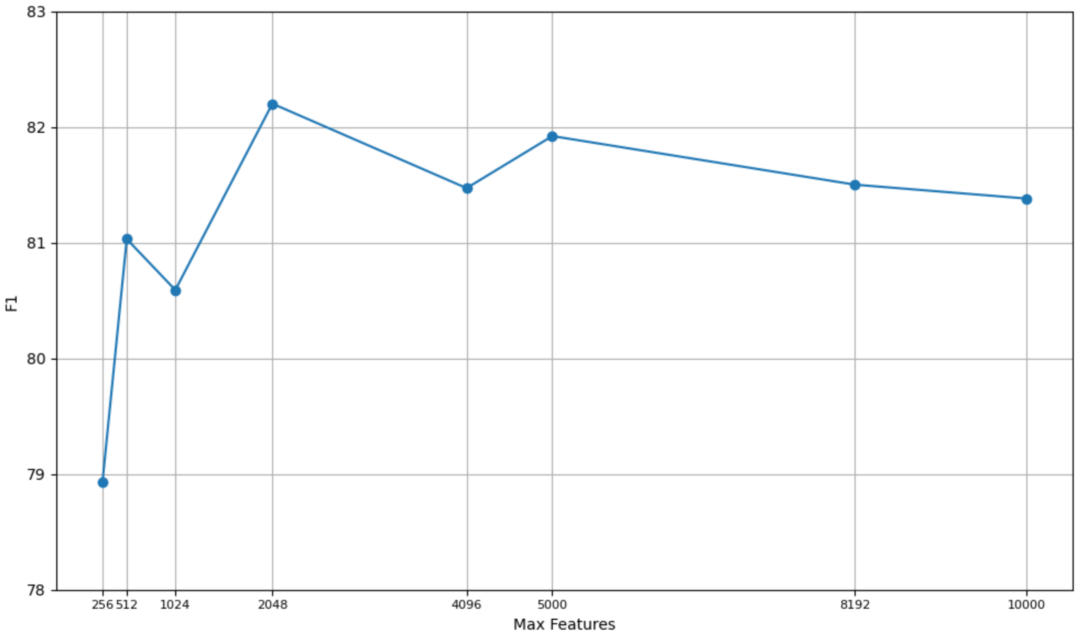


Figure 4.3 – Capturing the best BoW vector size for boosting the DNN performance.

The F1-score starts relatively high around 79.5% for 256 Max Features. As the number of Max Features increases to 512 and 1024, the F1-score drops sharply to around 80.5% and then 78.93% respectively. However, when the Max Features reaches 2048, the F1 score rises sharply to around 82.2%, indicating an improvement in performance. After the peak, the F1-score gradually declines as the Max Features increases further to 4096, 5000, and 10000, with the final value around 80.5% for 10000 Max Features. After analyzing all the results, the DNN model with 1-3 grams representation using BoW and 2048 vector size emerges as the optimal choice for our M2 model. It demonstrates a robust balance across precision, recall, and F1-score, making it the top candidate for M2 in our framework. An explained description of the model architecture is presented in Table 4.11.

Step	**Description**
Model Initialization	`Sequential` model.
Input Layer	Dense layer with 512 neurons and ReLU activation.
Hidden Layers	3 dense layers with 512 neurons and ReLU activation.
Dropout Layers	3 dropout layers with a rate of 0.5 after each dense layer.
Output Layer	Dense layer with `num_classes` neurons and softmax activation.
Model Compilation	Compile the model with categorical cross-entropy loss and Adam optimizer.
Model Training	Train the model with 100 epochs, batch size of 32, and validation data.

Table 4.11 – Description of the DNN Model Architecture.

4.4 Stacking based Detection

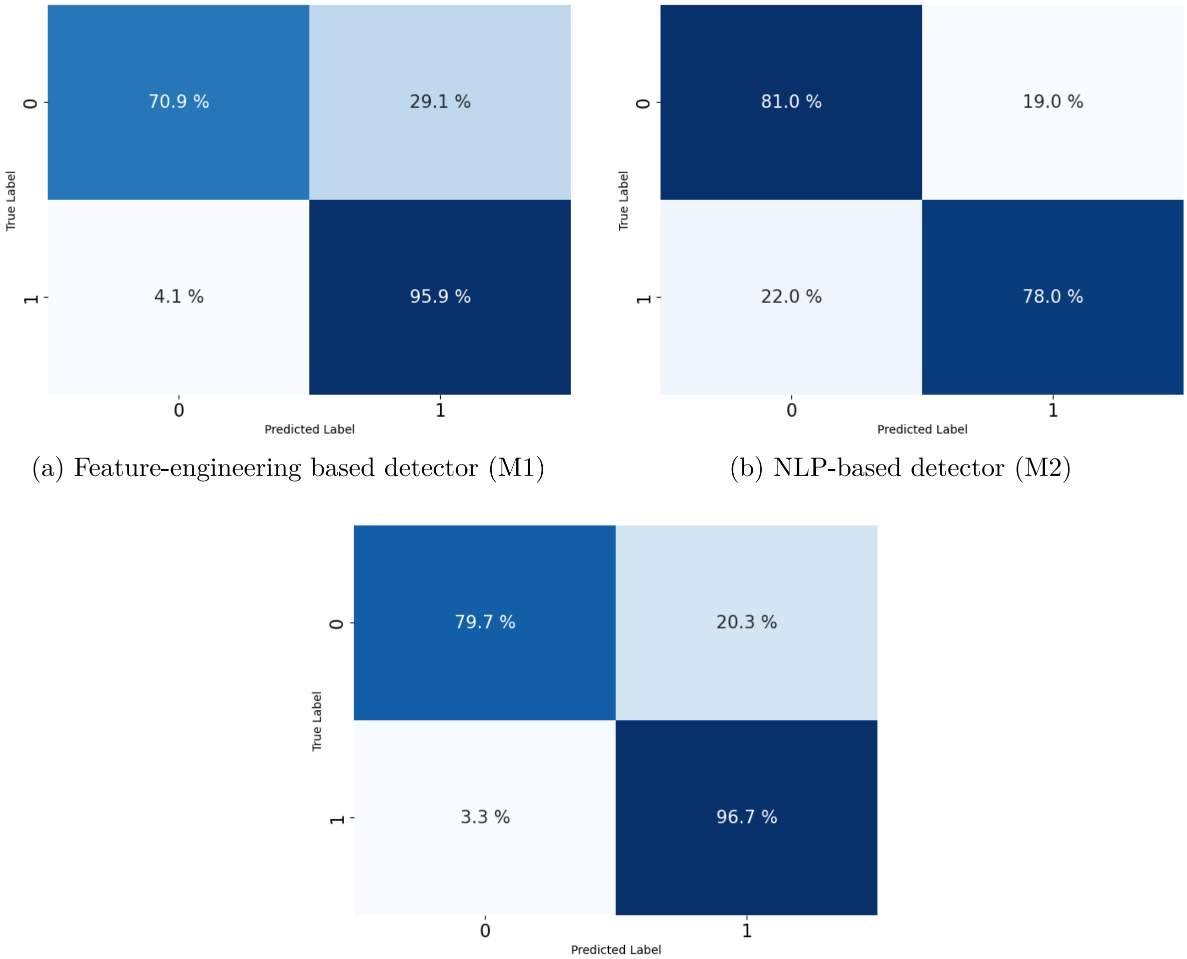
After selecting the best candidates for the M1 and M2 models from the previous experiments (GBM for M1 and DNN for M2), we initiated a stacking process to determine the optimal meta-model M3. We experimented with multiple models, and the results obtained are showcased in Table 4.12. This table presents the performance evaluation metrics for four different stacking models. The classifiers chosen as candidates for becoming meta-models are NB, LR, DT, and RF. The selection of these classifiers as meta-model candidates is based on their distinct characteristics and performance in various machine learning tasks. NB is chosen for its simplicity, computational efficiency, and effectiveness in handling high-dimensional data with categorical features. LR is a widely-used linear classification algorithm known for its interpretability, scalability, and robustness in binary classification tasks. DT offers intuitive interpretability, easy visualization, and the capability to capture complex relationships in the data. RF, an ensemble method built on the foundation of decision trees, provides robustness against overfitting, high accuracy, and the ability to handle large datasets with high dimensionality. By selecting a diverse set of meta-model candidates, we aim to explore different modeling approaches and harness the strengths of each algorithm to construct a powerful and versatile meta-model for our stacking ensemble. These are represented in Table 4.12 as *Stack-M3(M1,M2)*.

Model	Recall (%)	Precision (%)	Accuracy (%)	F1-score (%)
Stack-DT(GBM,DNN)	91.56	86.43	87.66	88.92
Stack-NB(GBM,DNN)	**96.90**	84.13	88.08	89.72
Stack-RF(GBM,DNN)	91.09	**88.60**	**88.86**	89.83
Stack-LR(GBM,DNN)	96.56	84.66	88.67	**90.22**

Table 4.12 – Performance of the stacking model

After analyzing the results, all models exhibited close performance, but *Stack-*

LR(GBM,DNN) demonstrates slightly superior performance, achieving a recall of 96.56%, precision of 84.66%, accuracy of 88.67%, and an F1-score of 90.22%. For more demonstration, we generated the confusion matrices for all the 3 models M1, M2 and M3 to display the impact of stacking on the performance of bot detection task.



(a) Feature-engineering based detector (M1)

(b) NLP-based detector (M2)

(c) Stacking-based detector (M3)

Table 4.13 – Confusion matrices for base and meta-detectors: M1, M2, and M3 models

The analysis of the confusion matrices for the base models (M1 and M2), as well as the meta-model (M3) shown in Figure 4.13, reveals that the stacking process significantly enhanced the performance of both the M1 and M2 models in detecting bots. For human detection, the M1 model achieved a detection rate of 70.9%, while the M2 model improved this detection rate to 81.0%. After applying stacking, the human detection rate reached 79.7% outperforming M1 and shows a slight decrease compared to M2. Regarding bot detection, the M1 model achieved a detection rate of 95.9%, and the M2 model achieved a detection rate of 78.0%. With stacking, this detection rate is increased to 96.7%. This improvement in bot detection is particularly significant, as the primary focus of this study is on detecting bots. The stacking process not only improved the overall detection rates but also demonstrated the effectiveness of combining multiple models to leverage their strengths and mitigate individual weaknesses.

4.4.1 Comparison with State-of-the-art methods

After analyzing the performance of various methods on the TwiBot-20 dataset, we compiled the results into Table 4.14. The table shows that our hybrid methodology achieved the highest accuracy and F1-score among all the methods tested, underscoring its effectiveness in bot detection on the TwiBot-20 dataset.

Work	**Classifier**	**Approach**	**Acc (%)**	**F1 (%)**
		TwiBot20 dataset		
Lee et al. [59]	RF	Hybrid	74.56	78.23
Yang et al. [94]	RF	Profile	81.91	85.46
Kudugunta et al. [56]	LSTM	Hybrid	81.74	75.17
Wei et al. [90]	Bi-LSTM	Content	81.74	75.17
Ours	Stack-LR (GBM, DNN)	Hybrid	**88.67**	**90.22**
		Cresci2015 dataset		
Prabhu Kavin et al. [13]	SVM	Content	/	93.11
Gao et al.[40]	bi-SN-LSTM	Content	**99.99**	**99.31**
Ours	DNN	Content	96.04	96.01
		Cresci2017 dataset		
Najari et al. [69]	GAN	Content	**94.90**	**95.80**
Heidari et al. [46]	LSTM	Content	94.60	94.10
Ours	DNN	Content	93.54	93.63

Table 4.14 – Comparison of various works on different datasets.

We extended our methodology to additional datasets, specifically *Cresci2017* and *Cresci2015*. Due to the limited availability of datasets suitable for our hybrid approach, we focused solely on evaluating the M2 model (DNN) rather than the entire framework. This limitation arose because many features utilized in our model are absent from most available datasets, often due to deprecation or other factors. Despite these constraints, our model demonstrated robust performance on both datasets. While it did not outperform all models presented in the table, several factors contribute to this outcome, including differences in the samples used to train the models.Nevertheless, our framework's performance was comparable to the models we benchmarked against, which achieved slightly better results. This results indicates that our methodology generalizes well across different data sources and is not confined to the TwiBot-20 dataset.

4.5 Conclusion

In this chapter, we presented a comprehensive experimental evaluation of our proposed hybrid ensemble methodology for detecting bots on Twitter. By leveraging the TwiBot-20 dataset, we were able to train and test our models on a diverse and representative collection of user data. The experimental results validated the effectiveness of our proposed methodology, demonstrating its ability to leverage complementary strengths of profile-based features and natural language processing for robust bot detection on Twitter. The stacking ensemble's superior performance underscored the merits of combining multiple models to mitigate individual weaknesses and harness collective strengths. Through rigorous feature analysis, we identified the most informative profile-based and content-based features for distinguishing bots from human accounts. Among the machine learning classifiers trained on engineered features, the Gradient Boosting emerged as the top performer to become our M1 model. For natural language processing techniques, we found that a deep neural network utilizing Bag-of-Words with an optimal 1-3 ram range and vector size of 2048 delivered the highest performance, becoming our M2 model. By stacking the predictions from M1 and M2 through a logistic regression meta-model M3, we attained substantial performance gains. The stacked ensemble significantly outperformed the individual base models across all evaluation metrics. Comparative analysis against state-of-the-art methods further highlighted the advantages of our hybrid approach. Finally, The models were evaluated on diverse datasets to ensure the generalizability of the framework. The experimental results validated the effectiveness of our proposed methodology, demonstrating its ability to leverage complementary strengths of profile-based features, content-based features and natural language processing for robust bot detection on Twitter. The stacking ensemble's superior performance underscored the merits of combining multiple models. Overall, this chapter provided empirical evidence supporting the potential of our hybrid ensemble framework in advancing the field of Twitter bot detection, paving the way for future research and practical applications in combating the growing threat of malicious bot activities on social media platforms.

Summary and conclusions

In today's digital era, social media bots pose a significant and growing threat to online social networks. These bots, capable of automating tasks such as posting, commenting, and interacting with users, can manipulate public opinion, spread misinformation, and conduct coordinated harassment campaigns. The increasing sophistication of social media bots makes distinguishing between genuine human interactions and automated activities challenging, thus jeopardizing the integrity of online discourse.

To address the issue of social media bot detection on Twitter, the work performed in this thesis proposes a robust hybrid ensemble approach that combines profile-based and content-based analysis. This approach leverages the strengths of both methods, enabling it to capture a wider range of bot behavior and characteristics, leading to more accurate and robust detection. To achieve this, we developed a framework incorporating two separate machine learning models. The first model is trained on features extracted through a feature-engineering technique from profile metadata and newly proposed features engineered from the content of tweets posted by each account. The second model is trained on features extracted through natural language processing (NLP) techniques applied to the posted tweets. Extensive analysis has been conducted to identify the best machine learning classifier for each approach. A stacking approach is then used for combining the individual models, and the best meta-classifier is also identified by exploring various machine learning classifiers. The proposed framework aims to effectively distinguish bots from genuine users or human accounts based on their profiles, activities, and interactions. This hybrid approach is designed to identify and mitigate the influence of social media bots on Twitter, providing a comprehensive solution for detecting and addressing these automated accounts. Experiments with the most challenging dataset (TwiBot20) demonstrate a significant improvement in performance of the proposed framework compared to state-of-the-art approaches tested on the same dataset. Additional experiments on other datasets further demonstrate the robustness and effectiveness of the proposed framework. This comprehensive evaluation underscores the potential of our hybrid approach in enhancing the detection and management of social media bots on Twitter.

While the proposed hybrid ensemble framework shows promise in identifying and mitigating the influence of socail media bots, it is not without limitations. One critique is the challenge of keeping up with the rapidly evolving tactics of bot developers. As detection methods improve, bot creators continuously adapt their strategies to evade detection, leading to an ongoing arms race. Additionally, the model's generalizability is a notable concern. While it shows effectiveness in detecting bots on Twitter, its applicability to other social media platforms, such as Facebook or Instagram, remains limited. This limitation arises because the proposed methodology primarily targets the unique characteristics and behaviors of Twitter bots, which may not be directly transferable to other platforms with different user dynamics and interaction patterns. Furthermore, the complexity and computational demands of the hybrid ensemble algorithms present practical challenges. Real-time implementation of such sophisticated algorithms requires substantial computational resources, which can be a significant barrier to scalability. This complexity can impede the deployment of the framework in real-world scenarios where immediate bot detection and response are crucial. Addressing these challenges is essential for the framework to achieve broader applicability and effectiveness across diverse social media platforms. Consequently, future work can focus on several key areas to enhance the proposed solution. One promising direction is to adapt the model for various social media platforms and emerging bot behaviors. Each platform has unique characteristics and user interaction patterns, so tailoring the model to platforms like Facebook, Instagram, and newer social media services will broaden its applicability and effectiveness. Additionally, staying abreast of the evolving tactics of bot developers is crucial, as it ensures the model remains robust against new and sophisticated bot strategies. Incorporating real-time detection capabilities is another vital area for improvement. Real-time detection would enable the timely identification and mitigation of bot activities, which is essential for minimizing their impact. Furthermore, integrating additional features could significantly improve detection performance. For instance, incorporating network-based metrics, such as the structure and dynamics of social connections, can provide deeper insights into bot behavior. Analyzing behavioral patterns over time, including posting frequency, content diversity, and interaction anomalies, can also enhance the model's ability to distinguish between genuine users and bots. Finally, addressing the computational challenges by leveraging distributed computing frameworks can make the solution more scalable and practical for large-scale applications. This will enable the framework to handle vast amounts of data typical in social media environments, ensuring it remains effective and responsive in real-world applications.

References

[1] Abokhodair, N., Yoo, D., and McDonald, D. W. Dissecting a social botnet: Growth, content and influence in twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (New York, NY, USA, 2015), CSCW '15, Association for Computing Machinery, p. 839851.

[2] Abreu, J. V. F., Ralha, C. G., and Gondim, J. J. C. Twitter bot detection with reduced feature set. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)* (2020), IEEE, pp. 1–6.

[3] Abubakar, H. D., Umar, M., and Bakale, M. A. Sentiment classification: Review of text vectorization methods: Bag of words, tf-idf, word2vec and doc2vec. *SLU Journal of Science and Technology 4*, 1 & 2 (2022), 27–33.

[4] Al-Qurishi, M., Al-Rakhami, M., Alamri, A., Alrubaian, M., Rahman, S. M. M., and Hossain, M. S. Sybil defense techniques in online social networks: a survey. *IEEE Access 5* (2017), 1200–1219.

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

[6] Alarifi, A., Alsaleh, M., and Al-Salman, A. Twitter turing test: Identifying social machines. *Information Sciences 372* (2016), 332–346.

[7] Alarifi, A., Alsaleh, M., and Al-Salman, A. Twitter turing test: Identifying social machines. *Information Sciences 372* (2016), 332–346.

[8] Alharthi, R., Alhothali, A., and Moria, K. Detecting and characterizing arab spammers campaigns in twitter. In *Proceedings of the 16th Learning and Technology Conference on Artificial Intelligence and Machine Learning* (2019), pp. 248–256.

[9] Alhassun, A. S., and Rassam, M. A. A combined text-based and metadata-based deep-learning framework for the detection of spam accounts on the social media platform twitter. *Processes 10*, 3 (2022), 439.

[10] Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., and Alomari, D. M. Machine learning-based social media bot detection: a comprehensive literature review. *Social Network Analysis and Mining 13*, 1 (2023), 20.

[11] Aral, S., and Walker, D. Tie strength, embeddedness & social influence: A large-scale networked experiment. *Management Science 60*, 6 (2013), 1352–1370.

[12] Auger, A., and Doerr, B. *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* Springer, Berlin, Germany, 2023.

[13] B, P. K., S, K., S, H., D, S., R, V., M, T., SLA, H., D, J., V, T., PR, K., and AG, A. Machine learning-based secure data acquisition for fake accounts detection in future mobile communication networks. *Wireless Communications and Mobile Computing* (2022).

[14] Barhate, S., Mangla, R., Panjwani, D., Gatkal, S., and Kazi, F. Twitter bot detection and their influence in hashtag manipulation. In *2020 IEEE 17th India Council International Conference (INDICON)* (2020), IEEE, pp. 1–7.

[15] Bird, S., Klein, E., and Loper, E. *Natural Language Processing with Python.* O'Reilly Media, Sebastopol, CA, 2009.

[16] Bishop, C. M. *Pattern Recognition and Machine Learning.* Springer, New York, NY, 2006.

[17] Boshmaf, Y., Logothetis, D., Siganos, G., Lería, J., Lorenzo, J., Ripeanu, M., Beznosov, K., and Halawa, H. Íntegro: Leveraging victim prediction for robust fake account detection in large scale osns. *Computers & Security 61* (2016), 142–168.

[18] Boshmaf, Y., Muslukhov, I., Beznosov, K., and Ripeanu, M. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference* (New York, NY, USA, 2011), ACSAC '11, Association for Computing Machinery, p. 93102.

[19] Breiman, L. Stacked regressions. *Machine learning 24*, 1 (1996), 49–64.

[20] Bruns, A., and Burgess, J. Researching news discussion on Twitter: New methodologies. *Journalism Studies 13*, 5-6 (2012), 801–814.

[21] Cable, J., and Hugh, G. Applying machine learning to identify social media trolls. Tech. rep., Stanford University, 2019.

[22] Castells, M. *Networks of outrage and hope: Social movements in the Internet age*. John Wiley & Sons, 2015.

[23] Collins, B. After mueller report, twitter bots pushed russiagate hoax narrative. https://www.nbcnews.com/tech/tech-news/after-mueller-report-twitter-bots-pushed-russiagate-hoax-narrative-n997441, 2019. Accessed: April 27, 2024.

[24] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems 80* (2015), 56–71.

[25] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. Fame for sale: efficient detection of fake twitter followers. *Decision Support Systems 80* (December 2015), 56–71.

[26] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. Dna-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems 31*, 5 (2016), 58–64.

[27] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion* (Republic and Canton of Geneva, CHE, 2017), WWW '17 Companion, International World Wide Web Conferences Steering Committee, p. 963972.

[28] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion* (2017), WWW '17 Companion, International World Wide Web Conferences Steering Committee, pp. 963–972.

[29] Dehghan, A., Siuta, K., Skorupka, A., Dubey, A., Betlen, A., Miller, D., Xu, W., Kamiski, B., and Praat, P. Detecting bots in social-networks using node and structural embeddings. *Journal of Big Data 10*, 1 (2023), 119.

[30] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[31] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*, 2nd ed. Wiley-Interscience, New York, NY, 2001.

[32] Echeverrï£¡ a, J., De Cristofaro, E., Kourtellis, N., Leontiadis, I., Stringhini, G., and Zhou, S. Lobo: Evaluation of generalization deficiencies in twitter bot classifiers. In *Proceedings of the 34th annual computer security applications conference* (2018), pp. 137–146.

[33] Efthimion, P. G., Payne, S., and Proferes, N. Supervised machine learning bot detection techniques to identify social twitter bots. *SMU Data Science Review 1*, 2 (2018), 5.

[34] Estellés-Arolas, E., and González-Ladrón-de Guevara, F. Towards an integrated crowdsourcing definition. *Journal of Information science 38*, 2 (2012), 189–200.

[35] Faerber, M., Qurdina, A., and Ahmedi, L. Identifying twitter bots using a convolutional neural network. In *proceedings of the 2019 Conference and Labs of the Evaluation Forum* (01 2019), pp. 1–8.

[36] Fan, Z., Gong, Y., Liu, D., Wei, Z., Wang, S., Jiao, J., Duan, N., Zhang, R., and Huang, X. Mask attention networks: Rethinking and strengthen transformer. *arXiv preprint arXiv:2103.13597* (2021).

[37] Ferrara, E. Social bot detection in the age of chatgpt: Challenges and opportunities. *First Monday 28*, 6 (2023).

[38] Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. The rise of social bots. *Communications of the ACM 59*, 7 (2016), 96–104.

[39] Freelon, D., McIlwain, C. D., and Clark, M. Beyond the hashtags:# ferguson,# blacklivesmatter, and the online struggle for offline justice. *Center for Media & Social Impact, American University, Forthcoming* (2016).

[40] Gao, T., Luo, X., Xia, Y., Xu, Z., Qiu, M., and Gao, J. A content-based method for sybil detection in online social networks via deep learning. *IEEE Access 8* (2020), 49302–49315.

[41] Garcia-Silva, A., Berrio, C., and Gomez-Perez, J. M. Understanding transformers for bot detection in twitter. *arXiv preprint arXiv:2104.06182* (2021).

[42] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning.* MIT Press, Cambridge, MA, 2016.

[43] Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, Burlington, MA, 2011.

[44] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, New York, NY, 2009.

[45] Hayawi, K., Saha, S., Masud, M. M., Mathew, S. S., and Kaosar, M. Social media bot detection with deep learning methods: a systematic review. *Neural Computing and Applications 35*, 12 (2023), 8903–8918.

[46] Heidari, M., Jones, J. H., and Uzuner, O. Deep contextualized word embedding for text-based online user profiling to detect social bots on twitter. In *2020 International Conference on Data Mining Workshops (ICDMW)* (2020), IEEE, pp. 480–487.

[47] Huberman, B. A., Romero, D. M., and Wu, F. Social networks that matter: Twitter under the microscope. *Information Systems: Behavioral & Social Methods* (2008).

[48] Irizarry, R. A. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R.* Chapman & Hall/CRC Data Science Series, 2020.

[49] Java, A., Song, X., Finin, T., and Tseng, B. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis* (8 2007), pp. 56–65.

[50] Jia, J., Wang, B., and Gong, N. Z. Random walk based fake account detection in online social networks. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (2017), pp. 273–284.

[51] Jurafsky, D., and Martin, J. H. *Speech and Language Processing*, 3rd ed. Pearson, Upper Saddle River, NJ, 2021.

[52] Kaplan, A. M., and Haenlein, M. Users of the world, unite! the challenges and opportunities of social media. *Business horizons 53*, 1 (2010), 59–68.

[53] Kemp, S. Digital 2023 global overview report. `https://datareportal.com/reports/digital-2023-global-overview-report`, 2023. Accessed: April 27, 2024.

[54] Khalil, H., Khan, M. U. S., and Ali, M. Feature selection for unsupervised bot detection. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies: Idea to Innovation for Building the Knowledge Economy, ICoMET 2020* (2020).

[55] Kietzmann, J. H., Hermkens, K., McCarthy, I. P., and Silvestre, B. S. Social media? get serious! understanding the functional building blocks of social media. *Business horizons 54*, 3 (2011), 241–251.

[56] Kudugunta, S., and Ferrara, E. Deep neural networks for bot detection. *Information Sciences 467* (2018), 312–322.

[57] Kwak, H., Lee, C., Park, H., and Moon, S. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web* (2010), pp. 591–600.

[58] Labrecque, L. I., Markos, E., and Milne, G. R. Online personal branding: Processes, challenges, and implications. *Journal of Interactive Marketing 25*, 1 (2011), 37–50.

[59] Lee, K., Eoff, B., and Caverlee, J. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (2011), ACM, pp. 185–192.

[60] Liang, W., Zhang, Y., Wu, Z., Lepp, H., Ji, W., Zhao, X., Cao, H., Liu, S., He, S., Huang, Z., et al. Mapping the increasing use of llms in scientific papers. *arXiv preprint arXiv:2404.01268* (2024).

[61] Liddy, E. D. *Natural language processing.* Encyclopedia of Library and Information Science, 2001.

[62] Lotan, G., Graeff, E., Ananny, M., Gaffney, D., Pearce, I., and Boyd, D. The Arab Spring | the revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International Journal of Communication 5* (2011), 31.

[63] Mangold, W. G., and Faulds, D. J. Social media: The new hybrid element of the promotion mix. *Business horizons 52*, 4 (2009), 357–365.

[64] Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval.* Cambridge University Press, Cambridge, England, 2008.

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

[66] Mitter, S., Wagner, C., and Strohmaier, M. A categorization scheme for socialbot attacks in online social networks. *arXiv preprint arXiv:1402.6288* (2014).

[67] Moore, A. *Machine Learning.* Prentice Hall, Upper Saddle River, NJ, 1997.

[68] Murphy, K. P. *Machine Learning: A Probabilistic Perspective.* MIT Press, Cambridge, MA, 2012.

[69] Najari, S., Salehi, M., and Farahbakhsh, R. Ganbot: A gan-based framework for social bot detection. *Social Network Analysis and Mining 12*, 1 (2022), 4.

[70] Orabi, M., Mouheb, D., Al Aghbari, Z., and Kamel, I. Detection of bots in social media: a systematic review. *Information Processing & Management 57*, 4 (2020), 102250.

[71] Papacharissi, Z. *Affective Publics: Sentiment, Technology, and Politics.* Oxford University Press, USA, 2015.

[72] Rajendran, G., Ram, A., Vijayan, V., and Poornachandran, P. Deep temporal analysis of twitter bots. In *Machine Learning and Metaheuristics Algorithms, and Applications: First Symposium, SoMMA 2019, Trivandrum, India, December 18–21, 2019, Revised Selected Papers 1* (2020), Springer, pp. 38–48.

[73] Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Flammini, A., and Menczer, F. Detecting and tracking political abuse in social media. In *Proceedings of the International AAAI Conference on Web and social media* (2011), pp. 297–304.

[74] Santhi, M., Latha, M., and Mrss, R. Impact of soical media and social media management tools for smart businesses. *International Journal of Advance Research and Innovative Ideas in Education 5*, 6 (2019), 1289–1295.

[75] Saranya Shree, S., Subhiksha, C., and Subhashini, R. Prediction of fake instagram profiles using machine learning. *Available at SSRN 3802584* (2021).

[76] Shafahi, M., Kempers, L., and Afsarmanesh, H. Phishing through social bots on twitter. In *2016 IEEE international conference on big data (big data)* (2016), IEEE, pp. 3703–3712.

[77] Shao, C., Ciampaglia, G. L., Varol, O., Yang, K.-C., Flammini, A., and Menczer, F. The spread of low-credibility content by social bots. *Nature communications 9*, 1 (2018), 1–9.

[78] Statista. Global social networks ranked by number of users. `https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users`. Accessed: April 27, 2024.

[79] Statista. Human and bot web traffic share worldwide from 2014 to 2022. https://www.statista.com/statistics/1264226/human-and-bot-web-traffic-share/, 2022. Accessed: April 27, 2024.

[80] Statista. Number of social media users worldwide from 2017 to 2027. https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users, 2024. Accessed: April 27, 2024.

[81] Stella, M., Ferrara, E., and Domenico, M. D. Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences 115*, 49 (2018), 12435–12440.

[82] Stieglitz, S., Brachten, F., Ross, B., and Jung, A.-K. Do social bots dream of electric sheep? a categorisation of social media bot accounts. *arXiv preprint arXiv:1710.04044* (2017).

[83] Suarez-Lledo, V., and Alvarez-Galvez, J. Assessing the role of social bots during the covid-19 pandemic: Infodemic, disagreement, and criticism. *Journal of Medical Internet Research 24*, 8 (8 2022), e36085.

[84] Tess, P. A. The role of social media in higher education classes (real and virtual)–a literature review. *Computers in human behavior 29*, 5 (2013), A60–A68.

[85] Twitter. Twitter usage statistics. https://about.twitter.com/en_us/company/brand-resources.html. Accessed: April 27, 2024.

[86] Twitter. Twitter api documentation. https://developer.x.com/en/docs/twitter-api, 2024.

[87] Van Engelen, J. E., and Hoos, H. H. A survey on semi-supervised learning. *Machine learning 109*, 2 (2020), 373–440.

[88] Veil, S. R., Buehner, T., and Palenchar, M. J. A work-in-process literature review: Incorporating social media in risk and crisis communication. *Journal of contingencies and crisis management 19*, 2 (2011), 110–122.

[89] Wang, G., Mohanlal, M., Wilson, C., Wang, X., Metzger, M., Zheng, H., and Zhao, B. Y. Social turing tests: Crowdsourcing sybil detection. *arXiv preprint arXiv:1205.3856* (2012).

[90] Wei, F., and Nguyen, U. T. 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)* (2019), IEEE, pp. 101–109.

[91] West, D. B., et al. *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001.

[92] Wilmott, P. Machine learning: an applied mathematics introduction. *Machine Learning and the City: Applications in Architecture and Urban Design* (2022), 217–248.

[93] Wu, W., Alvarez, J., Liu, C., and Sun, H.-M. Bot detection using unsupervised machine learning. *Microsystem Technologies 24* (01 2018).

[94] Yang, K.-C., Varol, O., Hui, P.-M., and Menczer, F. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence* (2020), pp. 1096–1103.

[95] Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD) 8*, 1 (2014), 1–29.

[96] Yu, T., and Zhu, H. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689* (2020).

[97] Zaefferer, M. Hyperparameter tuning approaches. In *Hyperparameter Tuning for Machine and Deep Learning with R*, E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, Eds. Springer, Singapore, 2023.

[98] Zeng, Z., Li, T., Sun, S., Sun, J., and Yin, J. A novel semi-supervised self-training method based on resampling for twitter fake account identification. *Data Technology and Applications 56*, 3 (2021), 409–428.

[99] Zhou, Z.-H., and Feng, J. Deep Forest: towards an alternative to deep neural networks. In *IJCAI* (2017), pp. 3553–3559.

[100] Zhou, Z.-H., and Feng, J. Deep forest. *National Science Review 6*, 1 (2019), 74–86.

[101] Zhu, X. J. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison, Madison, USA, 2005. Accessed: May 16, 2024.