United Arab Emirates University
Scholarworks@UAEU

Dissertations | Electronic Theses and Dissertations

6-2023

STREAM-EVOLVING BOT DETECTION FRAMEWORK USING GRAPH-BASED AND FEATURE-BASED APPROACHES FOR IDENTIFYING SOCIAL BOTS ON TWITTER

Eiman Alothali

Follow this and additional works at: https://scholarworks.uaeu.ac.ae/all_dissertations

Part of the Software Engineering Commons

UAEU

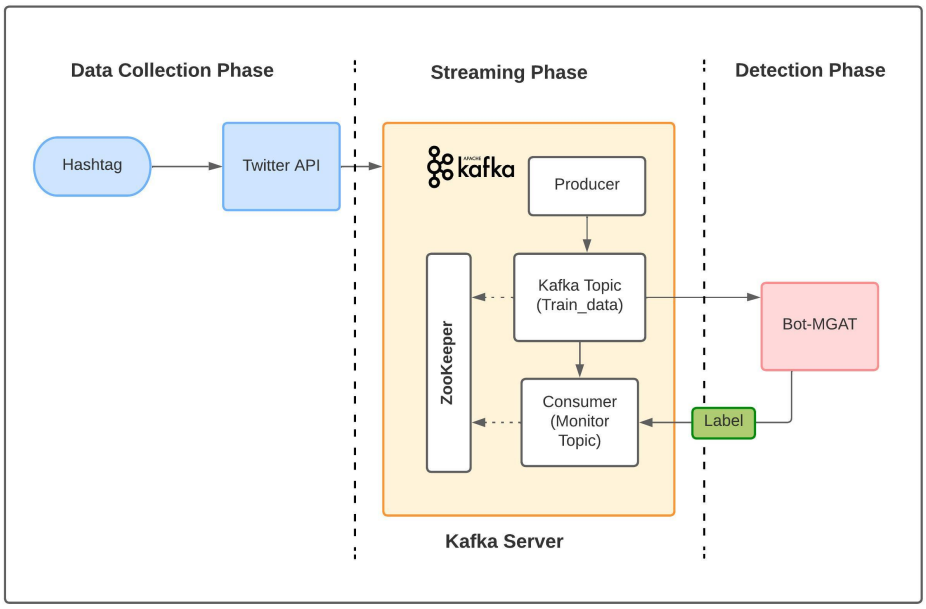
جامعة الإمارات العربية المتحدة
United Arab Emirates University

DOCTORATE DISSERTATION NO. 2023: 31

College of Information Technology

STREAM-EVOLVING BOT DETECTION FRAMEWORK USING GRAPH-BASED AND FEATURE-BASED APPROACHES FOR IDENTIFYING SOCIAL BOTS ON TWITTER

Eiman Alothali



June 2023

United Arab Emirates University

College of Information Technology

STREAM-EVOLVING BOT DETECTION FRAMEWORK USING GRAPH-BASED AND FEATURE-BASED APPROACHES FOR IDENTIFYING SOCIAL BOTS ON TWITTER

Eiman Alothali

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Informatics and Computing

June 2023

United Arab Emirates University Doctorate Dissertation
2023: 31

Cover: SEBD Framework Architecture

(Photo: By Eiman Alothali)

© 2023 Eiman Alothali, Al Ain, UAE
All Rights Reserved
Print: University Print Service, UAEU 2023

Declaration of Original Work

I, Eiman Alothali, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this dissertation entitled "*Stream-Evolving Bot Detection Framework Using Graph-based and Feature-based Approaches for Identifying Social Bots on Twitter*", hereby, solemnly declare this is the original research work done by me under the supervision of Dr. Hany Alashwal, in the College of Information Technology at UAEU. This work has not previously formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my dissertation have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this dissertation.

Student's Signature:

Date: 10-6-2023

Advisory Committee

1) Advisor: Hany Alashwal

Title: Associate Professor

Department of Computer Science and Software Engineering

College of Information Technology

2) Member: Nazar Zaki

Title: Professor

Department of Computer Science and Software Engineering

College of Information Technology

3) Member: Zouheir Trabelsi

Title: Professor

Department of Information Systems and Security

College of Information Technology

4) Co-advisor [External]: Kadhim Hayawi

Title: Assistant Professor

Department of Computational Systems

College of Interdisciplinary Studies, Zayed University, UAE

Approval of the Doctorate Dissertation

This Doctorate Dissertation is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Hany Al Ashwal
   Title: Associate Professor
   Department of Computer Science and Software Engineering
   College of Information Technology
   Signature _____ Date 13/06/2023

2) Member: Fady Alnajjar
   Title: Associate Professor
   Department of Computer Science and Software Engineering
   College of Information Technology
   Signature _____ Date 13/6/2023

3) Member: Amir Ahmad
   Title: Associate Professor
   Department of Information Systems and Security
   College of Information Technology
   Signature _____ Date 13/06/2023

4) Member: Suhaila Binti Zainudin (External Examiner):
   Title: Associate Professor
   Centre for Artificial Intelligence Technology (CAIT), Faculty of Information Science and Technology
   Institution: Universiti Kebangsaan Malaysia
   Signature _____ Date 14 June 2023

This Doctorate Dissertation is accepted by:

OBO/

Dean of the College of Information Technology: Professor Taieb Znati

Signature _____

Date 23/08/2023

Dean of the College of Graduate Studies: Professor Ali Al-Marzouqi

Signature Ali Hassan

Date 23/08/2023

Abstract

This dissertation focuses on the problem of evolving social bots in online social networks, particularly Twitter. Such accounts spread misinformation and inflate social network content to mislead the masses. The main objective of this dissertation is to propose a stream-based evolving bot detection framework (SEBD), which was constructed using both graph- and feature-based models. It was built using Python, a real-time streaming engine (Apache Kafka version 3.2), and our pretrained model (bot multi-view graph attention network (Bot-MGAT)). The feature-based model was used to identify predictive features for bot detection and evaluate the SEBD predictions. The graph-based model was used to facilitate multiview graph attention networks (GATs) with fellowship links to build our framework for predicting account labels from streams. A probably approximately correct learning framework was applied to confirm the accuracy and confidence levels of SEBD.

The results showed that the SEBD can effectively identify bots from streams and profile features are sufficient for detecting social bots. The pretrained Bot-MGAT model uses fellowship links to reveal hidden information that can aid in identifying bot accounts. The significant contributions of this study are the development of a stream-based bot detection framework for detecting social bots based on a given hashtag and the proposal of a hybrid approach for feature selection to identify predictive features for identifying bot accounts. Our findings indicate that Twitter has a higher percentage of active bots than humans in hashtags. The results indicated that stream-based detection is more effective than offline detection by achieving accuracy score 96.9%. Finally, semi-supervised learning (SSL) can solve the issue of labeled data in bot detection tasks.

Keywords: Social Bots, Detection, Real-Time, Stream, Machine Learning, GAT, PAC, SL, SSL, Twitter.

Title and Abstract (in Arabic)

إطار بث لاكتشاف الروبوتات المتطورة باستخدام النماذج القائمة على الرسم البياني والميزات لتحديد الروبوتات الاجتماعية على شبكة تويتر

الملخص

تركز هذه الرسالة على مشكلة تطور حسابات الروبوتات الاجتماعية في شبكات التواصل الاجتماعي عبر الانترنت ، وخاصة شبكة تويتر. حيث تنشر مثل هذه الحسابات معلومات مضللة وتضخم محتوى الشبكة الاجتماعية لتضليل الجماهير. الهدف الرئيسي من هذه الرسالة هو اقتراح إطار متطور لاكتشاف الروبوتات (SEBD)، والذي تم إنشاؤه باستخدام كل من النماذج القائمة على الرسم البياني والميزات. تم بناؤه باستخدام لغة البرمجة (Python) ، ومحرك دفق في الوقت الفعلي (Apache Kafka version 3.2). ونموذجنا الذي تم اختباره مسبقاً؛ شبكة الانتباه للرسم البياني متعدد العروض (Bot-MGAT). تم استخدام النموذج القائم على الميزات لتحديد الميزات التنبؤية لاكتشاف الروبوتات وتقييم تنبؤات (SEBD). تم استخدام النموذج القائم على الرسم البياني لتسهيل استخدام شبكات الاهتمام بالرسم البياني متعددة العروض (GATs) مع روابط الزمالة لبناء إطار للتنبؤ بنوع الحساب من التدفقات. تم تطبيق إطار تعليمي تقريبي لتأكيد مستويات الدقة والثقة في الإطار المقترح (SEBD).

أظهرت النتائج أن (SEBD) يمكنه تحديد الروبوتات بشكل فعال من التدفقات، وأن ميزات الملف الشخصي كافية لاكتشاف حسابات الروبوت الاجتماعية. كما أظهرت النتائج أن نموذج (Bot-MGAT) الذي تم اختباره مسبقاً من خلال روابط الزمالة يستطيع الكشف عن المعلومات المخفية من خلال هذه الروابط للمساعدة على تحديد حسابات الروبوت. تتمثل المساهمات المهمة لهذه الدراسة في تطوير إطار عمل لاكتشاف الروبوتات القائمة على التدفق لاكتشاف برامج الروبوت الاجتماعية بناءً على علامة تصنيف معينة (Hashtag) واقتراح نهج مختلط لاختيار الميزات لتحديد الميزات التنبؤية لتحديد هذه الحسابات الروبوتية. تشير النتائج التي توصلنا إليها أن شبكة التواصل تويتر تحتوي على أعداد كثير من الروبوتات النشطة بنسبة أعلى من البشر في علامات التصنيف (Hashtags). أشارت النتائج إلى أن الاكتشاف المستند إلى التدفق أكثر فعالية من الاكتشاف في الوضع التقليدي بنسبة دقة 96.9%. أخيراً ، يمكن للتعلم شبه الخاضع للإشراف (SSL) حل مشكلة نقص البيانات المصنفة للقيام بمهام اكتشاف حسابات الروبوت.

مفاهيم البحث الرئيسية: برامج الروبوت الاجتماعية، الاكتشاف، الوقت الفعلي، البث ، التعلم الألي، شبكات الاهتمام بالرسم البياني، الشبكات العصبية للرسم البياني، التعلم الصحيح التقريبي، التعلم الخاضع للإشراف، شبكة تويتر.

List of Publications

This dissertation is based on the work presented in the following papers:

I. Eiman Alothali, Kadhim Hayawi, Hany Alashwal. Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter. Social Network Analysis and Mining 2021;11(1):1-15.

II. Eiman Alothali, Motamen Salih, Kadhim Hayawi, Hany Alashwal. Bot-MGAT: A Transfer Learning Model Based on a Multi-View Graph Attention Network to Detect Social Bots. Applied Sciences 2022;12(16):8117.

III. Eiman Alothali, Kadhim Hayawi, Hany Alashwal. SEBD: A Stream Evolving Bot detection framework with application of PAC learning approach to maintain accuracy and confidence levels. Applied Sciences 2023;13(7):4443.

Author's Contribution

The contribution of Eiman Alothali to the papers included in this dissertation was as follows:

I. Sole responsibility for planning the research, conducting the experiments, and preparing the manuscript.

II. Sole responsibility for the data collection and processing, evaluation of results and manuscript writing.

III. Addressing the reviewers' comments on the reviewed manuscripts.

Author Profile

Eiman Alothali received her bachelor's degree in Information Technology from Zayed University, Abu Dhabi; UAE. She joined Abu Dhabi Education Knowledge Department as a computer teacher and worked as a teacher in different schools in Abu Dhabi. She received her Master in Information Technology in Cyber Security in 2014 from Zayed University, Abu Dhabi. She was awarded her highness Sheikha Fatima bint Mubarak award for honor students. Her research interest is in the fields of cyber security, machine learning, and social networks analysis. In 2017, she was awarded SMBZ grant from Emirates Foundation for Youth to be able to join the Computer Science Ph.D. program at the UAEU. Currently, she mainly focuses on using advanced machine learning techniques for social network analysis. She has more than five years of research experience. During her PhD studies she participated in different conferences and journal publications.

Acknowledgements

All grace and thanks first are to Allah, the Almighty, for giving me the continual motivation, patience, and abilities to complete my PhD degree.

My profound gratitude is to my advisor, Dr. Hany Alashwal, for his constant support, guidance, and encouragement during my PhD journey. I am honored to be his first PhD candidate at UAEU. His knowledge, patience, and constructive feedback have been important in shaping my experience as a researcher and producing this PhD thesis.

I would also like to thank my co-advisor, Dr. Kadhim Hayawi, for his guidance, insights, and commitment to my academic development. In which, I was able to develop my ideas and generate high-quality research with his informative remarks and suggestions.

I would like to specially thank Prof. Nazar Zaki and Prof. Zouheir Trabelsi, members of my advisory committee, for their insightful comments and constructive input on my research. Their unique viewpoints and knowledge have improved my work and broadened my intellectual horizons.

I am also grateful to Ms. Maryam Al Mandhri for her constant support and assistance, my colleagues in CIT, and along with the Emirates Foundation for Youth for its SMBZ grant for my PhD degree.

Finally, I would like to express my deepest appreciation to my family especially my Mom and Dad, my Sisters (Mariam, Elham, Manal), and my best friend Najla for their encouragement, and unwavering support throughout my PhD journey. This accomplishment would not have been possible without their support.

Thank you all.

Dedication

To my beloved parents and family; especially my nephew Abdullah

Table of Contents

Title i
Declaration of Original Work iii
Advisory Committee iv
Approval of the Doctorate Dissertation v
Abstract vii
Title and Abstract (in Arabic) viii
List of Publications ix
Author's Contribution x
Author Profile xi
Acknowledgements xii
Dedication xiii
Table of Contents xiv
List of Tables xvi
List of Figures xvii
List of Abbreviations xviii
Chapter 1: Introduction 1
1.1 Social Networks Attacks 3
1.2 Motivation 4
1.3 Problem Statement 5
1.4 Research Objectives 6
1.5 Literature Review 7
1.5.1 Feature-based Methods 11
1.5.2 Graph–based Methods 17
1.5.3 Crowdsourcing 20
1.6 Research Gaps 22
Chapter 2: Methods and Results 25
2.1 Research Methodology 25
2.2 Data Collection and Preprocessing 28
2.3 Model Evaluation 30
2.4 Experimental Environment 31

2.5 Objective 1: To Develop an Effective Method for Evaluating and Extracting Features and Analyzing Bot Behavior to Improve Social Bot Detection in Real-time 31
2.5.1 Objective 1: Method 33
2.5.2 Objective 1: Results 34
2.6 Objective 2: To Develop a Machine Learning Models, Including Feature-Based and Graph-Based Models, to Enhance Accuracy in Real-time 38
2.6.1 Objective 2: Method 38
2.6.2 Objective 2: Result 42
2.7 Objective 3: To Develop a Method Capable of Adapting to the Dynamic Nature of Social Networks and the Complex Changes in Bot Features Over Time 50
2.7.1 Objective 3: Method 51
2.7.2 Objective 3: Result 52
Chapter 3: Findings & Discussion 56
3.1 Dataset Size 56
3.2 Profile Features for Detecting Social Bots 57
3.3 Stream Bot Detection 60
3.4 Handling the Dynamic Nature of Social Networks 64
3.5 Research Limitations 65
Chapter 4: Conclusions and Future Perspectives 67
4.1 Research Implications 67
4.2 Future Work 68
References 69
List of Other Publications 79
Appendix 80

List of Tables

Table 1.1: Summary of literature on social bot detection on Twitter 9
Table 1.2: Strengths and limitations of social bot detection approaches 23
Table 1.3: SEBD and existing frameworks/tools for social bot detection 24
Table 2.1: Statistics of the used datasets 29
Table 2.2: Datasets that were retrieved and processed by SEBD 29
Table 2.3: Performance for feature subset selection for the bot class 37
Table 2.4: Results of SEBD and other baselines 44
Table 2.5: Botometer results using group 1 dataset 44
Table 2.6: Result of benchmark dataset with RF train/test using group 1 dataset 45
Table 2.7: Over-fitting techniques on Bot-MGAT 48
Table 2.8: Performances of Bot-MGAT and baselines 53
Table 2.9: Sample size ranges with minimum and maximum FNRs and FPRs for the group 2 dataset with PAC parameters 54
Table 3.1: Descriptions of features selected for SEBD 57

List of Figures

Figure 1.1: Illustration of a social bot network in an OSN 2
Figure 1.2: Types of OSN attacks 4
Figure 1.3: Research questions 6
Figure 2.1: Overview of the research methodology 27
Figure 2.2: Categories of Twitter features 32
Figure 2.3: Algorithm for finding predictive features 34
Figure 2.4: Results of the proposed feature selection methods 35
Figure 2.5: Precision vs Recall curve for CVEA method 36
Figure 2.6: SEBD framework 39
Figure 2.7: Bot-MGAT Model with Transfer learning 40
Figure 2.8: SEBD flowchart 41
Figure 2.9: SEBD perfromance results (A) SEBD confusion matrix
(B) SEBD results for the holdout method and cross-validation
with RF 43
Figure 2.9: ROC curve using Botometer with CAP threshold 0.5 44
Figure 2.10: ROC curve using Botometer with CAP threshold 0.75 45
Figure 2.11: ROC curve for benchmark dataset using train/test with
group 1 dataset 46
Figure 2.12: Results of SEBD with different SL classifiers 47
Figure 2.13: ROC curves of the SEBD framework with different SL
classifiers 48
Figure 2.14: Bot-MGAT without regularization for validation loss and
training loss 49
Figure 2.15: Effect of regularization L1 on Bot-MGAT 49
Figure 2.16: Effect of regularization L2 on Bot-MGAT 50
Figure 2.17: Transfer learning approach 52
Figure 2.18: Empirical analysis using PAC framework for SEBD 54
Figure 2.19: Model accuracy according to the number of estimators 55
Figure 3.1: Frequency distributions of the verified feature across all datasets 59
Figure 3.2: Screen name lengths and account ages in the Twibot20, Kaggle,
Github, and SEBD datasets 60
Figure 3.3: Visualization of SEBD results using t-SNE 61
Figure 3.4: AUCs of undersampling algorithms for the SEBD dataset 63
Figure 3.5: AUCs of undersampling algorithms for the Twibot-20 dataset 63
Figure 3.6: AUCs of undersampling algorithms for the Kaggle dataset 64
Figure 3.7: AUCs of undersampling algorithms for the Github dataset 64

List of Abbreviations

API	Application Programming Interface
CGN	Convolutional Graph Network
CNN	Convolutional Neural Network
DL	Deep Learning
FNR	False Negative Rate
FPR	False Positive Rate
GAT	Graph Attention Networks
GNN	Graph Neural Network
KNN	K-Nearest Neighbor
LR	Logistic Regression
MVGAT	Multi-View Graph Attention Networks
NB	Naïve Byes
NN	Neural Network
OSNs	Online Social Networks
PAC	Probability Approximately Correct Learning
RF	Random Forest
RGCN	Relational Graph Convolutional Network
ROC	Receiver Operating Characteristic Curve
SEBD	Stream-Evolving Bot Detection
SL	Supervised Learning
SSL	Semi-Supervised Learning
SVM	Support Vector Machine
UL	Unsupervised Learning

Chapter 1: Introduction

Online social networks (OSNs) serve as a universal platform for sharing and promoting businesses, links, viewpoints, and news. Users can share content and links via social networks' data-sharing features; however, this functionality can be exploited by scammers and individuals with illicit intentions. OSNs are exposed to threats due to social bot accounts. Social bots are programs that automatically generate content, distribute it via a particular social network, and interact with its users [1–3]. Newer generations of bots are more sophisticated than the older generations and tougher to detect [4]. Between 9% and 15% of Twitter accounts are operated by bots, representing the equivalent of 48 million accounts [5, 6]. One investigation revealed that 35% of the information posted on Twitter had originally been shared by social bots [7]. A recent study found that 83.1% of the tweets on climate change discussions on Twitter were generated by social bots [8].

Early research sought to address the issues related to automated social media accounts' ability to propagate spam, phishing URL links, or manipulate genuine accounts by stealing users' identities and misleading individuals [9–12], as well as their use in general [13–15]. Malicious accounts frequently work with a botmaster (Figure 1.1), who manages a team of social bots using Command & Control (C&C) attack to disperse spam or influence user behavior on a certain social network [16]. For example, in Syria, a social bot was employed to flood Twitter with hashtags related to the Syrian Civil War and irrelevant topics that redirected the attention of users from controversial government actions [7]. Social bots have also played significant roles in uprisings occurring in the aftermath of major events such as elections or conflicts [1, 17–21]. Gupta et al. [22] studied the fake content proliferated via Twitter during the Boston Marathon blasts in 2013 and its role in spreading rumors and misinformation. They found that bot accounts were generated after the blasts, and most of them impersonated real accounts. The malicious activities of bots during such events can be used to spread spam. In addition, they can also cause financial harm, as observed in the case of CYNK Technology in 2014, whose market price fell by a factor of 220 due to social bots engaging in automated stock trading [2]. Even after the changes in Twitter policy regarding

verification [23], a bot account recently impersonated the insulin company Eli Lilly and caused a 4.37% drop in its stock price [24]. A study in 2019 by Cresci et al. [25] found that 71% of stock tweets related to five main financial markets in the US were generated by a group of botnets. In addition to the financial content, sophisticated bots and Russian trolls have been responsible for contaminating health-related content and amplifying vaccine debate messages that can result in adverse health implications for Twitter's users [16].

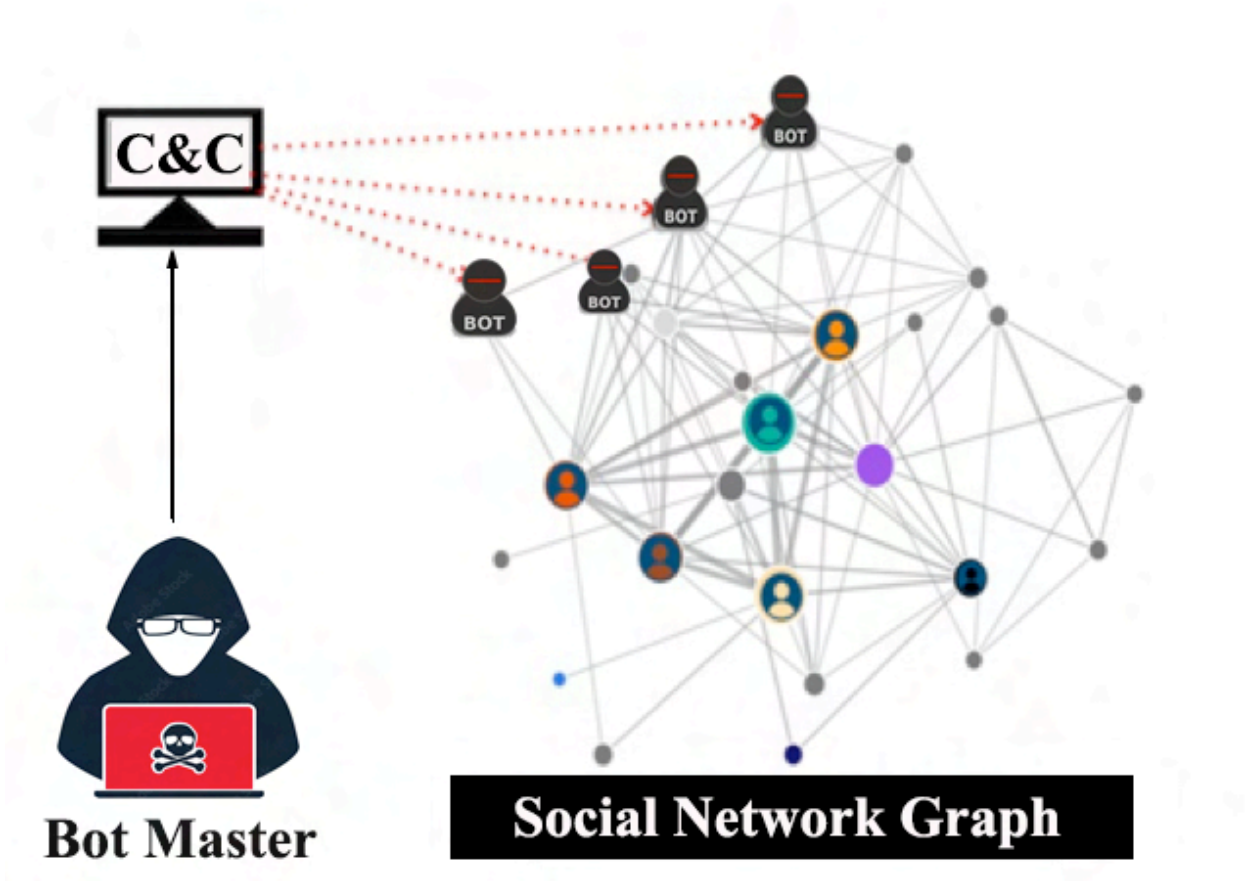


Figure 1.1: Illustration of a social bot network in an OSN

In recent years, the world has witnessed the interference of political bots in various elections. The goal was to promote influential, polarized, and misleading content, such as in the 2016 US elections [8–10], 2017 German elections [11, 12], 2017 Ecuador elections [13], and Russian regime party orientation [14]. In [11], the authors analyzed German communities that generated the content during the 2017 German elections and found that bots constituted 11% of the communities. Due to the many

artificial social connections evident, social bot activity also affects OSNs' social graph. Social bots have the ability to gather private information from user accounts and utilize it for phishing and spamming operations [12, 26]. Additionally, they can gather data from the internet to adopt other people's personas, imitate their habits, and persuade people by ranking and retweeting. Social bots can harm the social network's environment in addition to fundamentally fooling users by creating fictitious fellowship relations [27] and/or contaminating the network's content.

Scholars have suggested various strategies for detecting and preventing social bots in an effort to reduce the risks that they pose. Most research in this area to date has centered on examining behavioral patterns [28–32]. For instance, a study by Fu et al. [33] offered a dynamic meter to assess the change in users' behaviors as a means of pinpointing the tactics spammers utilize. Another detection method attempted to locate rogue account groups by comprehending the methods underlying the created account names and subsequently connecting them to the moment of creation [34]. This study has an acceptable degree of accuracy as it spanned 4.7 million Twitter accounts. Stringhini et al. [11] conducted research in the same area using data from three significant OSNs after building extensive and varied honey profiles. They successfully detected 15,857 spam profiles that were deleted by Twitter.

1.1 Social Networks Attacks

The popularity and widespread use of OSNs for connecting people to share interests and personal information has allowed intruders and attackers to gain easy access to user information, resulting in privacy and security issues. Security concerns have arisen as OSN websites have become targets of different types of attacks. These attacks can be classified into three categories: classic, modern, and adolescent [35] (See Figure 1.2). Classic attacks, which involve installing viruses or malicious code in a user’s browser or profile, have been occurring since the early days of the Internet. In contrast, the recently evolved modern and adolescent attacks are pervasive threats that have emerged with the rise of OSNs.

Social bot attacks involve the use of bots (botnets) that are created and controlled by a botmaster to execute malicious behavior and simultaneously mimic the

communication of ordinary OSN users to minimize the probability of being detected [35]. In this study, we focus on this type of OSN attack and develop a detection system to identify accounts with malicious intentions.

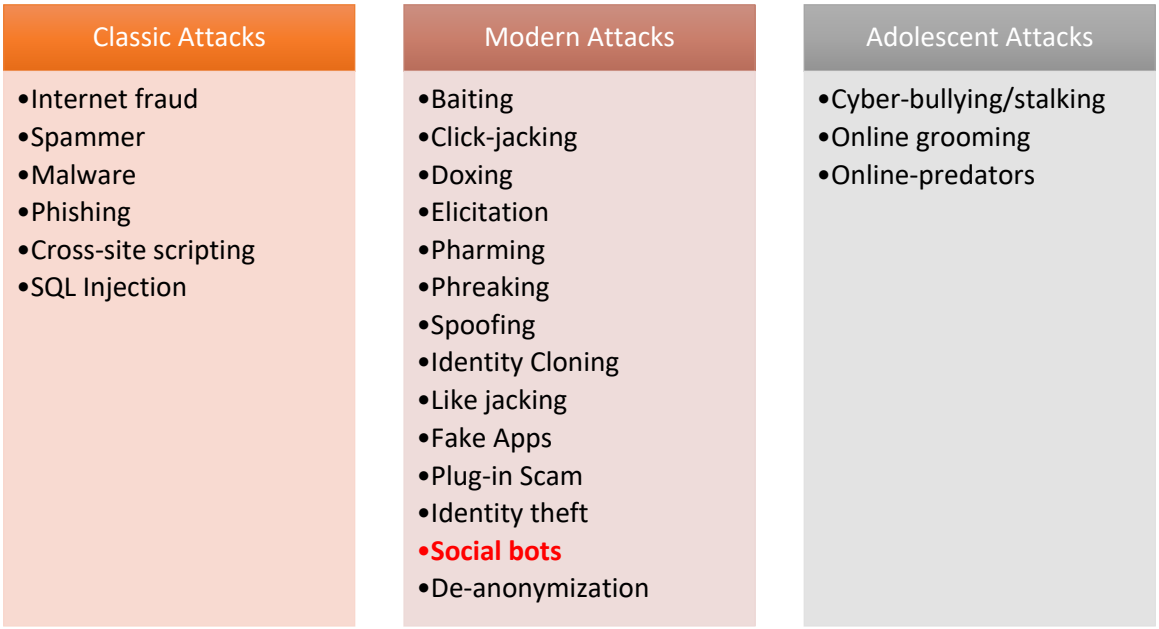


Figure 1.2: Types of OSN attacks

1.2 Motivation

The power of OSNs, such as Twitter, has been demonstrated worldwide, from the user to the government level. They offer an opportunity for the whole world to participate in discussions on any topics of interest (such as football) and to witness current events in real-time (such as the coronavirus pandemic). Hence, malicious actors have targeted OSNs to spread rumors and misinformation [2, 16, 36, 37]. Therefore, the impact of using social bots that impersonate human users on OSNs to attack users or communities by spreading false information, emphasizing polarization, or spreading hate speech can be considered a cyber threat to the stability of countries. Such threats can be identified by the impact they can have on people and their reactions. For example, consider the public response to selling or buying stocks based on fake financial stock market news spread and inflated by social bots on OSNs. Such behavior can affect the economies of countries and stability in other domains such as politics and trade [25]. Therefore, social bots can affect any domain [38].

Therefore, this research aims to raise awareness among OSN users about the critical role of social bots in the online space. It also aims to contribute to the research community by proposing solutions to this problem through research and investigations.

1.3 Problem Statement

In today's age of social media dominance, social bots continue to present a significant threat to the safety and credibility of online users. It can be challenging to identify social bots that represent serious hazards to finances, relationships, political stability, and security. The evolution of these bots' designs and capabilities to evade detection techniques has made risk evaluation and management a complex research problem [7, 39]. A study by [40] found only 38 out of every 120 social bots were identified and removed by Twitter. It's crucial to remember that not all social bot accounts can be categorized as dangerous accounts. Some even make it clear in the account profile that they are not operated by real people. Social bots that do not operate with harmful intent can perform a variety of beneficial functions, like curating news feeds or responding to customer service requests. However, in the current study, we were concerned with social bots that their intention is to inflate hashtags with certain content.

While existing approaches for detecting social bots have been evaluated, the available solutions based on selected features and trained classifiers have proven to have limitations in their accuracy and detection capabilities. As a result, there is a pressing need to develop a real-time detection framework that can overcome these limitations and identify social bots more effectively. One major challenge in detecting social bots is the ever-changing nature of their behavior. For example, an approach focused on identifying social bots based on their screen names is no longer effective, as bots have evolved to use more sophisticated methods to create believable names. Similarly, detecting automation behavior that can result in suspension of accounts by Twitter is also becoming increasingly difficult. To address these challenges, our proposed real-time detection framework will leverage the latest advances in machine learning. By constantly analyzing data and adapting to new tactics used by social bots, our solution will be able to more accurately and consistently identify and flag fraudulent accounts. Ultimately, this will help to improve the safety and security of online communities, while

safeguarding the integrity of social media platforms.

The proposed research aims to address the necessity of real-time bot detection and analysis of the existing social bot detection techniques used on the Twitter social network platform. Traditional offline detection techniques have shown a significant decrease in performance when tested on new data due to the adaptive nature of social bot accounts [41]. Furthermore, with the regular changes to Twitter API default values, adaptive social bot detection is crucial to keep up with these alterations. Detecting social bots in real-time when they are attacking is crucial when it comes to mitigating their impact, as opposed to post-incident detection, which reduces the impact. Therefore, this research aims to answer the following research questions as depicted in Figure 1.3.

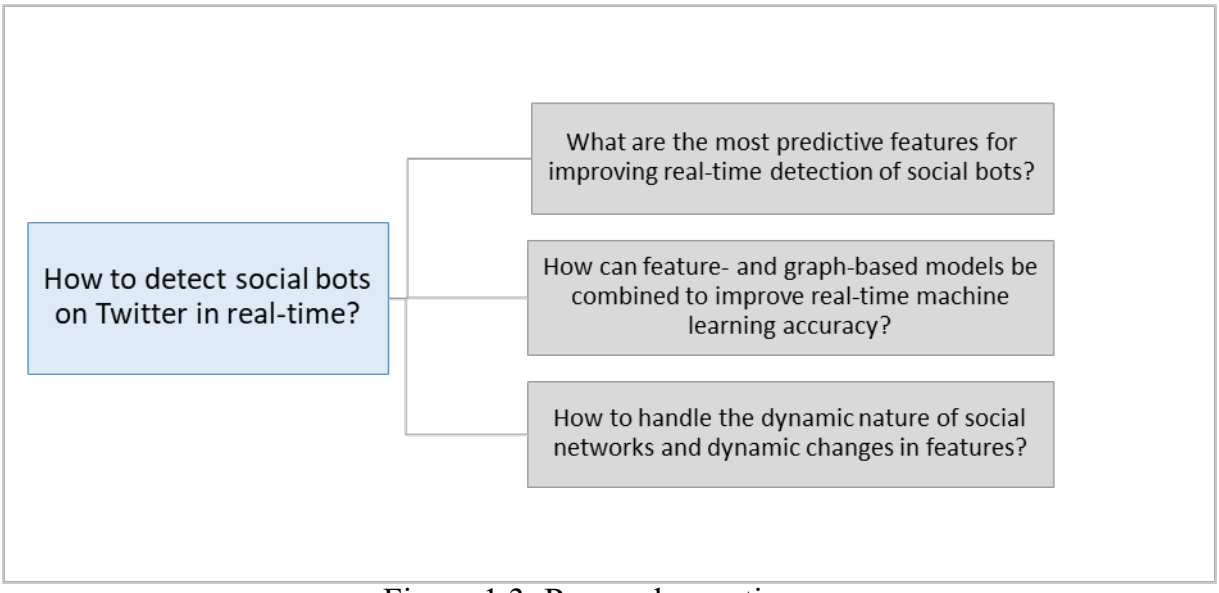


Figure 1.3: Research questions

1.4 Research Objectives

The major objective of the study is to provide a model that supports the real-time detection of social bots on OSNs, particularly the Twitter platform. The act of distinguishing between social bots and real accounts in their evolving attacks on trending hashtags is known as social bot detection. The main aims of this study, is to offer a framework that will facilitate social bot detection systems to overcome their current limitations. The objectives of this study are as follows:

1. To develop an effective method for evaluating and extracting features and analyzing bot behavior to improve social bot detection in real-time.
2. To develop a machine learning models, including feature-based and graph-based models, to enhance accuracy in real-time.
3. To develop a method capable of adapting to the dynamic nature of social networks and the complex changes in bot features over time.

The outcome of this comprehensive and thorough study is poised to yield significant benefits not only to the rapidly expanding field of cyber security but also to users and developers of social networks across various demographics. Furthermore, the insights gleaned from this research hold tremendous potential for the academic research community, particularly given the persistent and pressing nature of the problem at hand.

1.5 Literature Review

The implications and influence of social bots on OSN environments have been studied over the past few years. The sophisticated and deceptive nature of social bot accounts that adapt to behave like humans has made their detection challenging. Therefore, the methodologies that support social bot detection strategies can differ significantly. In general, nevertheless, they can be divided into three categories: feature-based, graph-based, and crowdsourcing [2, 42]. The first step in identifying social bots is to retrieve data from the Twitter feed. After the data has been gathered, the next step is to prepare it for the classifier of choice by removing and choosing the features that may be investigated using statistical approaches, as in [43, 44] or those that can be manually labeled using prior work, as in [9, 45, 46].

The feature-based approach involves developing algorithms and statistical methods that can help understand the revealing features or behaviors of social network accounts to distinguish between human- and computer-led activities. Graph-based approaches use the social graph of a social network to understand the network information and relationships between edges or links across accounts to detect bot activity. Finally, crowdsourcing involves using expert annotators to identify, evaluate, and determine social bot behavior through scientific measures. Recently, other methods have been proposed to detect botnets using generative models [47–49]. Using such

models is effective when the target class features are separable, because they generate synthetic data points to replicate the positive class for building the model. However, new social bot account generation techniques adapt to these features to avoid detection; therefore, features of such account are very close to those of the human class if tested using new datasets. Additionally, these computations are expensive for bot detection in a stream. However, using such a model to classify bot and human accounts based on text behavior has been proven to be effective [47].

Table 1.1 presents a summary of the three popular techniques researchers have used to identify social bot accounts on Twitter. The following sub-sections will present an overview of the relevant studies and datasets, the included classifiers, the detection procedures, and how the results were validated.

Table 1.1: Summary of literature on social bot detection on Twitter

Reference	Year	Method	Real-time Detection	Features						Dataset	Accuracy (%)	Other Measures
				Network	Profile	Content	Temporal	Sentiment	Other			
Rodrigues et al. [50]	2022	Feature-based	✓					✓		UCI	98.74	✓
Najari et al. [47]	2022	Generative model						✓		[51, 52]	98	
Shevtsov et al. [53]	2022	Feature-based		✓	✓	✓	✓	✓		P*	97.7	✓
Feng et al. [54]	2022	Graph-based		✓	✓	✓				[55]	86.64	✓
Mazza et al. [56]	2022	Graph-based			✓	✓				B		✓
Hayawi et al. [57]	2022	Feature-based			✓					B	97	✓
Yang et al. [58]	2022	Graph-based			✓	✓				B	87.84	✓
Feng et al. [59]	2021	Graph-based		✓	✓	✓				[55]	84.62	✓
Cui et al. [60]	2021	Graph-based						✓		O	97.7	
Feng et al. [61]	2021	Feature-based		✓	✓	✓				[55]	84	✓
Shukla et al. [62]	2021	Feature-based			✓					[63]	84	
Huang et al. [64]	2020	Graph-based				✓				O	92.4	
Sayyadiharikandeh et al. [65]	2020	Feature-based		✓	✓	✓	✓	✓		B	96	✓
Mendoza et al. [66]	2020	Graph-based		✓	✓	✓				[67]	93.86	✓
Rodríguez-Ruiz et al. [68]	2020	Feature-based			✓	✓				[52]	90	
Barhate et al. [69]	2020	Feature-based			✓					[41]	96	✓
Wu et al. [49]	2020	Generative model		✓	✓	✓		✓	✓	[5, 70]	97	✓
Alhosseini et al. [71]	2019	Graph-based		✓	✓					[72]	94	✓
Badawy et al. [73]	2019	Feature-based		✓	✓	✓	✓	✓	✓	P +USC	96	
Stukalet al. [74]	2019	Feature-based			✓	✓		✓		P	90	✓
Cresci et al. [25]	2019	Feature-based				✓	✓			P		✓
Loyola-González et al. [75]	2019	Feature-based			✓	✓	✓	✓		P + [52]	93.8	✓
Mazza et al. [67]	2019	Feature-based					✓		✓	P		✓
Hafzullah & Taner [76]	2019	Feature-based		✓	✓					P*	96.81	✓
Schuchard et al. [77]	2019	Graph-based		✓			✓			P		
Nazeret et al. [78]	2019	Feature-based				✓				[79]	≥ 84.9	✓
Gilani et al. [80]	2019	Crowdsourcing		✓		✓	✓			P		
Balestrucci et al. [81]	2019	Feature-based				✓	✓			[5]	82.26	

Table 1.1: Summary of literature on social bot detection on Twitter (Continued)

Reference	Year	Method	Real-Time Detection	Features						Dataset	Accuracy (%)	Other Measures
				Network	Profile	Content	Temporal	Sentiment	Other			
Albadi et al. [82]	2019	Crowdsourcing/Feature-based		✓	✓	✓	✓	✓		P*		✓
Cresci et al. [83]	2019	Feature-based				✓	✓			[52]	50.5	✓
Beskow&Carley [84]	2019	Feature-based			✓					P	99.6	✓
Kudugunta & Ferrara [51]	2018	Features-based			✓	✓		✓		[52]	>96	✓
De Cristofaro et al. [85]	2018	Feature-based			✓	✓				[52][86][9]+O	97	✓
Efthimion et al. [87]	2018	Features-based			✓	✓	✓	✓		[52] + N	97.75	✓
Barbon et al.[88]	2018	Features-based					✓	✓		P	94.47	
Cornelissen et al. [89]	2018	Graph-based		✓						[5]	70	✓
Stella et al. [90]	2018	Graph-based		✓		✓				P		
Gilani et al. [91]	2017	Crowdsourcing/ Feature-based			✓	✓	✓			[86]	86.44	✓
Cai et al. [92]	2017	Feature-based				✓	✓			P + [79]	87.32	✓
Kantepe & Ganiz[45]	2017	Feature-based			✓	✓	✓			P	86	✓
Minich et al. [93]	2017	Feature-based	✓	✓	✓	✓	✓			P*		✓
Jia et al. [94]	2017	Graph-based		✓						[95]	96	✓
Chavoshi et al. [96]	2016	Feature-based	✓		✓	✓	✓			P	94	✓
Mehrotra et al. [97]	2016	Graph-based		✓					✓	[27]	95	✓
Zhang et al. [98]	2016	Graph-based		✓						P		✓
Alarifi et al. [44]	2016	Crowdsourcing/Feature-based				✓	✓			P + [99]	88	✓
Chu et al. [100]	2012	Feature-based			✓	✓	✓			P	96	✓

P: Private data collected by authors (* these data are shared by authors)
UCI: https://archive.ics.uci.edu/ml/index.php
B: Botometer Repository (https://botometer.iuni.iu.edu/bot- repository/)
N: NBCNews from: https://www.nbcnews.com
O: Other datasets from previous studies
USC: From https://www.recode.net/2017/11/2/16598312/russia-twitter-trump-twitter-deactivated-handle-list

1.5.1 Feature-based Methods

ML techniques are commonly applied in feature-based approaches. These techniques help predict account types based on the selected predictive features of the user profile metadata, tweet metadata, temporal patterns, or a combination of all. A recent study [53] proposed a labeling pipeline wherein two off-the-shelf ML bot detection tools were used with a majority voting mechanism to maintain similar labels. Thereafter, a train and split with cross-validation was applied to evaluate the bot detection performance of the model, which achieved an area under the receiver operating characteristic (ROC) curve (AUC) and F1 scores of 97.7% and 89.6%, respectively. Another study [76] focused on profile and interaction features to classify the collected Twitter accounts as popular-active, observer-passive, or spam-bot malicious. They used the K-nearest neighbor (KNN), support vector machine (SVM), and artificial neural network (ANN) classification algorithms to build their model and achieved an accuracy of 96.81% with a recall rate ≥91% and precision ≥88%. In [100] they studied automation tweeting behavior, tweet content, and account features to spot bot. Authors analyzed the features to divide accounts into three categories: human, bot, and cyborg. They integrated random forest classifier for predictions, a bayesian classification to detect text patterns and an entropy-based aspect to measure automation. To evaluate the effectiveness of the method, they used a confusion matrix and achieved an average score of accuracy of 96%.

Social bot detection using deep learning algorithms has proven to be effective when considering data labeling and model specifications [50, 51, 74, 92]. A recent study [57] proposed a deep neural network with feature engineering to detect bot accounts. Their method uses the long short-term memory (LSTM) network and pretrained global vectors (GLoVe) to learn features, and achieved and AUC score of 97%. In [51], the authors proposed using a deep learning network based on contextual LSTM architecture for exploiting account metadata and content to detect bot accounts at the tweet level. The goal of limiting the feature size is to improve performance efficiency and interpretability. Their model achieved classification accuracies of 96 and 99% at the tweet and account metadata levels, respectively. It is important to note that the data used for training and testing in [52] were collected in 2016. In [74], deep learning was used to identify the

political orientation of Russian Twitter bots: pro-regime, anti-regime, and neutral. Their detection approach was based on a voting ensemble wherein the training component included four classifiers: SVM, ridge regression, extreme gradient boosting tree, and adaptive boosting (AdaBoost). Their model achieved an accuracy of 90%. In [92], a bot detection method based on a behavior-enhanced deep learning model for OSNs was proposed, using convolutional neural network (CNN) to capture the latent temporal patterns of users' tweet history and behavior. They applied LSTM long-short-term memory and hidden layers. This model achieved recall, precision, and F1 score of 86.26, 88.41, and 87.32%, respectively. Similarly, a recent study [50] proposed a real-time framework for analyzing Twitter sentiments using deep learning and ML for bot detection. They aimed to classify a tweet as spam or not using a semi-supervised approach. For this task, their results showed that the multinomial naïve Bayes classifier and LSTM model achieved accuracies of 97.78 and 98.74%, respectively.

Botometer, previously known as BotOrNot [43] is one of the earliest and most widely used bot detection systems. Their classifier uses RF, an ensemble supervised learning (SL) method, to evaluate social bots. This classification system extracts more than 1,000 features from six main classes: network, user, friends, temporal, content, and sentiment data. They used ten-fold cross-validation to measure their system performance and achieved an AUC of 95%. However, when the authors tested their system on new data, it achieved an accuracy of 85% [5]. In [65], the authors proposed an ensemble classification voting scheme inspired by the observation that human accounts are more homogeneous than bot accounts. Using a dataset from a previous study, their model achieved an AUC of 96% and was in good agreement with other baselines.

Kantepe and Ganiz proposed a framework based on SL for detecting bots on Twitter [45]. They collected data of 1,800 Twitter accounts and extracted 62 features that were categorized into three levels: user, tweet, and temporal. They used four classifiers: logistic regression, multinomial naïve Bayes, SVM, and gradient boosted trees; gradient boosted trees achieved the highest accuracy of 86% and an F1 score of 83%. Similarly, Gilani et al. [91] used a classification model to classify Twitter accounts into automated agents and human users. They performed three sets of experiments, through which they ran a five-fold cross-validation through training and testing. Their model achieved an

accuracy of 86.44%, precision of 85.4%, recall of 82.2%, and F-score of 83%. They found that 6 features scored the highest among the 15 features. A study [69] also compared the performance of supervised and unsupervised learning (UL) for detecting bot accounts for a public dataset, and the RF classifier obtained an AUC of 96%. Their findings indicate that bots have a lower follower growth rate than normal accounts. Additionally, they discovered that bot estimations in the studied hashtags ranged between 30 and 72%.

To identify Twitter users that are active for political debates or spreading news, both user and tweet metadata should be used to improve predictions. The experiment in [73] aimed to identify predictive features with the highest accuracy to determine the features for identifying users with maximum participation in spreading malicious content during the 2016 US elections. They classified the data features into five categories: metadata, linguistic inquiry and word count, engagement, activity, and other variables. They found that bot scores and political ideology features were the most predictive features for distinguishing users who spread malicious content. Additionally, the gradient boosting classifier performed better than other classifiers such as extra trees, RF, and AdaBoost.

In [81], a decision model using BotOrNot [43] was presented to identify credulous accounts exposed to bot activities. Once a user is identified as a human, an extended analysis process is performed to label their followers as humans or bots. After several analytical processes, the user account is categorized as a credulous Twitter user. Similarly, in [68] authors proposed one-class classification using some profile and content features to identify genuine classes. They aimed to discover bot accounts using an anomaly detection mechanism to identify behavior variances, as bots are not homogeneous.

Efthimion et al. [87] proposed an SL technique that uses an SVM to detect social bots on Twitter. They used publicly available datasets to analyze account profile metadata based on 15 variable assumptions. For example, they assumed that if an account has fewer than 30 followers, it is a bot account. They achieved an accuracy of 97.75% with a 2.25% misclassification rate. However, these assumptions might not

work, as sophisticated social bots keep adapting to mimic human behavior to avoid detection.

In [84], the authors proposed a toolbox approach that provided distinct models for detecting bots at four tiers of data granularity: semantics, account metadata, temporal patterns, and network patterns. Their work was based on random string classification, which was applied to Tier 1: the screen name of a Twitter account. They developed a random string detection model to detect randomly generated screen names using 15 random characters. Their model achieved an accuracy, F1 score, precision, and recall of 99.6, 99.6, 99.4, and 99.8%, respectively. Although this model can detect bot accounts that use random screen names, it is important to consider that some real users might also use random string screen names generated by Twitter. In addition, some recent bot accounts have started avoiding randomizing screen names and are increasingly using realistic names.

To detect social bot patterns by examining content, a contrast-classification pattern-based model was proposed in [75]. The same dataset as that in [52] was used to train the classifier on collective data for testing in Spanish. The authors combined features from [52] with the features proposed for their model. RF and contrast pattern-based classifiers achieved the best results among all classifiers, with an accuracy of 90%. They found that the human class has more patterns than the bot class; this should be considered to improve bot detection techniques based on behavior patterns. Barbon et al. [88] used text mining to understand the posting patterns and distinct writing styles of OSN users. They proposed a discrete wavelet transform-based algorithm to classify text authors into humans, legitimate bots, and malicious bots. They used the RF algorithm to induce a classification model into a wavelet model with two Twitter datasets. Their model achieved an accuracy of 94.47%.

In [82], a regression model was proposed to identify Arabic bots by studying hate speech on Arabic Twitter. They created a ground truth and trained their model after a manual labeling process to predict the bot scores. They used 35 features, including content, tweets, accounts, and sentiments, to train their regression model, and performed a statistical evaluation. They achieved a Spearman's rho of 0.74, Kendall's tau of 0.60,

and mean absolute error of 0.69. They found that bots generated 11% of hateful Arabic tweets in their dataset. Albadi et al. found that Botometer does not support Arabic content features and tends to assign higher bot scores to human accounts that use third-party feeds for Quran verses and to those that used their account for a short period of time, with less than 200 tweets in their timeline.

Nazer et al. proposed a bot detection model that focuses on recall without sacrificing overall performance [78]. They used Latent Dirichlet allocation to extract features from user tweet content and considered each user as a document and their tweets as topics. Their model achieved an accuracy of ≥84.9% and recall of ≥97.1%. In contrast, leave one botnet out (LOBO) [85] is an evaluation methodology for generalizing deficiencies in bot classifiers to enhance bot detection. They used a large and most varied collection of bots from literature to train their classifier, which achieved an accuracy of over 97%.

In [25], a large-scale analysis of bot and spam activity regarding stock markets on Twitter was presented. They employed a general digital DNA social bot detection method proposed in literature [70, 101] to collect data. This approach involves encoding a user's online behavior, such as tweets or retweets, into string characters to represent their digital DNA (DDNA), and then comparing multiple DDNA sequences of the studied group using bioinformatics algorithms. The system detected evolving bots with an F1 score of 0.97. The results showed that 71% of users that retweeted were classified as bots.

Using a genetic algorithm based on DDNA to simulate the evolution of spambot behavior, a synthetic evolving bot (GenBot) was proposed in [83]. They aimed to demonstrate the next-generation evolution of social bots to create effective detection techniques in the future. They compared non-evolved social bots with those created using their simulator. The accuracy dropped from 92.9% for non-evolved spambots to 50.5% for evolved spambots. This perspective is important to consider as it may offer malicious actors insights for adding more sophistication to current social bot designs to bypass detection.

An unsupervised group-based clustering approach, called retweet buster (RTBust), was proposed in [67]. The goal was to detect the malicious retweeting patterns of bots by implementing an LSTM network to extract patterns. Thereafter, density-based clustering is used to cluster users based on their latent features using an LSTM encoder. RTBust obtained an F1 score of 0.89 and detected two active botnets when it was applied to a large dataset of retweets.

Another tool that uses UL is DeBot [96]. It is a bot detection system that uses a pairwise approach. The authors applied a lag-sensitive hashing technique to cluster user accounts into correlated sets in real-time. They aimed to identify similar profiles based on real-time temporal behaviors, and employed dynamic time warping to capture the correlations between posting activities using a cross correlation-based random projection technique. They achieved 94% precision for their generated daily reports. Similarly, Minich et al. [93] proposed the BotWalk algorithm to explore bot accounts in a seed bank. Their framework used a near-real-time adaptive anomaly detection approach. They used a combination of depth- and breadth-first explorations to identify users' followers who are bot accounts and add them to the seed bots. The framework achieved a precision of 90%.

1.5.1.1 Strengths

One of the strengths of feature-based approaches for bot detection is their performance. However, this depends on aspects other than the selected predictive features, such as the training data, new test data, sample size, and selected classifier. For example, the performance of deep learning for bot detection with less features achieved accuracy almost 99% at the profile level and 96% at the tweet level [51]. However, with the rapid evolution of social bots, the accuracy decreases. Therefore, ML models should regularly adapt to the evolving patterns of social bots that can bypass detection, as discussed in [83]. Otherwise, they may not be able to accurately predict new samples.

1.5.1.2 Limitations

In feature-based approaches, identifying the most predictive features is crucial, and a model's inability to identify them can generate incorrect prediction results. Having a list of features to select from and combining other features, as in [73] can help

overcome this issue. Therefore, detecting social bots involves recognizing a range of predictive features. The classifier should be selected and trained based on features and data size. Because many studies have used RF or SVM classifiers owing to their good performance [3, 41, 78], the results vary, and other factors must be addressed. Fewer samples can cause the model to overpredict certain classes [74] and produce inaccurate results. Data size is an issue in feature-based approaches for detecting social bots. For instance, some classifiers perform better on small data sizes, whereas others require large sample sizes to produce reliable predictions.

The availability of labeled data is another issue in feature-based SL approaches as dataset sharing is constrained by Twitter user privacy terms. Thus, the number of features in available datasets vary to avoid violating user privacy agreements. In addition, many studies use the few available labeled datasets that were collected between 2015 and 2017 as ground truth datasets owing to the number of features they contain.

1.5.2 Graph–based Methods

Understanding social network graphs to distinguish user relationships on social networking platforms is a common approach. Trust propagation, graph clustering, and graph metrics are the three primary social graph-based techniques frequently used to identify rogue accounts and social bots [42]. A recent study [56] proposed an UL approach that involved dimensionality reduction techniques, such as uniform manifold approximation and projection, and density- and hierarchical-based clustering (HDBSCAN) to identify trolls, bots, and humans. They found that their bot detection method was more effective in identifying groups of accounts of the same type, with a cluster purity exceeding 0.9. Cornelissen et al. proposed a graph-based detection approach for identifying automated accounts using social network topology as a classifier [89]. They achieved a maximum accuracy of 70%. Therefore, graph-based approaches should be combined with other methods to obtain superior accuracy.

Several studies have integrated graph neural networks to detect social bots. Additionally, several studies have used heterogeneous information networks to model social graphs [54, 58, 60, 64]. In [54], the authors proposed a heterogeneity-aware bot detection method that leveraged the network topology and different relations between

OSN users. Their model achieved an accuracy of 86.64% and F1 score of 88.21%. Their results indicated that using only the graph structure reduced performance; therefore, graph models should also consider user features. In [65], the authors proposed a bot detection method that uses the PyTorch-BigGraph (PBG) tool for graph representations and joint interactions between users. Subsequently, label propagation is used to label nodes according to their proximities. Their model outperformed when both social network graphs and interaction links were used, compared to when only social network graphs were used. They also found that bots interacted more with certain human groups.

Similarly, Yang et al. [58] proposed a reinforced self-supervised graph neural network (GNN) framework. Their model determined the best multi-hop neighborhood and the number of layers in the GNN architecture. Recent studies have proposed employing graph convolutional neural networks (GCNNs) [71] and relational graph convolution networks (BotRGCN) [59] to improve spam bot detection. In [71], the authors used GCNNs on a well-known spam bot dataset, and achieved better results than those of previous methods. However, old datasets are easy to identify because they do not contain the most recent bots.

A study on the detection of social bots that inflate opposition on Twitter was conducted in [90]. They used a graph-based approach, implemented cutting-edge techniques, multilanguage sentiment analysis, network partitioning, and semantic networks to identify polarized groups. They used the PageRank centrality algorithm to identify human and bots accounts, and found that the number of average human users tended to be almost 1.8 times more central than bots in terms of information flow. However, bots tend to target humans by supporting instability- and hatred-related content.

In [77], the authors proposed a statistical network analysis framework for social bots. The goal was to characterize the relative influence of conversations and persistence ranking of social bots in OSNs. They used six centrality measurements to analyze the temporal persistence of the bots' centrality rankings. They found that bots made more attempts, on average, than humans to initiate communication through retweets. Social bots exhibited temporal persistence in the centrality ranking of OSN conversations.

SybilWalk [94] is a sybil (fake account) detection method that employs a random-walk-based method on an undirected social graph. The goal of the random walk approach is to assign benignness scores to trustworthy users while assigning negative scores to Sybil users. As a result, these scores will aid in dividing accounts into two groups: genuine and Sybil. Additionally, this technique can rank all users to help find the highest-ranked accounts that are most likely to be Sybils. The authors of SybilWalk made the assumption that the two linked nodes frequently had the same label. Using a Twitter directed graph dataset that they had gathered from a prior study [95], they labeled the genuine node badness score with 0 and 1 for the Sybil node. They achieved an AUC score of 0.96.

Mehrotra et al. proposed a technique to detect fake followers based on social graph-based attributes that are related to the centrality of each node in the graph [97]. They used six features from graph-based centrality measures to achieve the objective of classifying the centralities. They used three classifiers after calculating the centrality measurements of each given node in the graph. The random forest classifier achieved the maximum accuracy of 95%, along with precision and recall of 88.99% and 100%, respectively.

TrueTop[98] represents another impact measurement system that is based on graph-based methodology. To comply with the terms of service of the platform, the writers used a synthetic simulation of Twitter users. They utilized four datasets to create the system, assess the model's correctness, and gauge Sybil's resilience using a set of predefined metrics. They provided a model of the power of Sybil assaults based on the parameter, which was the ratio of the edge's total weight in the Sybil region to that in the non-Sybil zone. They found the worst-case scenario of Sybil assaults, where there was no interaction between the two zones.

1.5.2.1 Strengths

Recent studies have shown that GNN models are effective in terms of node classification and graph structures. Extracting information hidden between graph nodes by using node embeddings for classification tasks is a strength of GNN models. This is because graph-based models can represent complex relationships between entities, such

as in social networks. These models are flexible and interpretable for a wide range of applications such as classification, link prediction, and clustering[102–104].

Another advantage of graph-based methods is the detection of groups on OSNs [90]. Centrality ranking algorithms, such as PageRank, have achieved efficient processing times compared with the data size to measure persistence or influence [77]. In a graph-based approach, the visualization of network density via illustrations is helpful for a better understanding, as in [97, 98] which measured the impact rate of a social network graph based on the distance measured from the centrality of the highly weighted nodes.

1.5.2.2 Limitations

Graph-based techniques require making careful decisions when selecting metrics and assumptions to ensure scalability. The exact network may not be feasible for implementation in a real analysis owing to computational costs. Therefore, addressing such measurements is important to create a smaller network for analysis. The authors of [94] writers investigated how trust spreads in a network where user ranking is facilitated by trust ratings. Nevertheless, this method is sensitive to the choice of trusted seeds and relies on frequently unreliable assumptions in order to function. However, if the targeted network graph is significant and real, the computation cost may be considerable. Consequently, even if the outcomes in the real network can occasionally be unpredictable, using this strategy using a synthetic network is helpful.

1.5.3 Crowdsourcing

This approach of social bot detection involves using human skills to identify patterns across account profiles or the content shared by human and social bot accounts. The human expert's responsibility is to differentiate between automated accounts and human accounts.

A large-scale study based on a human annotation task was proposed in [80]. The authors performed a comparative analysis to determine the impact of bots and characterize the relationships between bots and humans on Twitter. They used the principal analysis component to evaluate their 22 feature selections and achieved a satisfaction result of nearly 100%. They discovered several trends between humans and

bots. For example, bots retweeted more often and posted five times as many URLs as humans in their tweets. In a previous work [91], they used group of annotators to classify accounts into bot or human. They have to provide a justification for their choice. They were provided with a list of features to consider, such as creation date of account, etc. To maintain reliability of results, the application of Cohen's kappa (k) coefficient and average pairwise agreement was performed.

One of the first annotations used to detect social bots was organized by DARPA as a Twitter bot detection challenge competition in 2015 [9]. The competition involved identifying influential bots that supported pro-vaccination discussions on Twitter to serve as ground truth. All teams used human judgment to identify bots after using their own implementations of bot detection techniques based on the features they selected to inform their guesses. Similarly, in [44], 10 volunteers rated and labeled 2000 random accounts to build a ground truth dataset. They evaluated the accuracy and reliability of their labeling process by injecting their ground truth dataset into a subset of 1,020 Twitter bot accounts by using another ground truth reference. They achieved an accuracy of 96%, with a 4% error rate for the labeling process.

1.5.3.1 Strengths

Human intelligence is effective in detecting straightforward abnormalities. For example, some of the content circulated by social bots circulate is generated by ignoring language features. Therefore, bot accounts can be easily identified if they retweet multiple times in five different languages. Human annotators can identify this anomaly, whereas ML approaches might not.

1.5.3.2 Limitations

Although we trust the human ability to identify differences and similarities, human errors inevitably occur during annotation tasks [44]. This approach involves manual processing, which requires time and effort. The lack of automation tools to assist with annotation makes it a challenging task. In [80], the authors used an annotation task wherein "bot" and "human" ground-truth labels were assigned to the dataset and the annotations were compared against an online bot detection tool for evaluation for a more reliable annotation classification.

1.6 Research Gaps

A summary of the strengths and limitations of each approach is presented in Table 1.2. Identifying bot accounts is a challenging task; however, this study relies on the detection of social bots while they participate in an evolving hashtag. As presented in the literature review in Section 1.4, most social bot detection techniques are based on offline processing, wherein data are collected, manually labeled, and then processed to identify social bot accounts. This study makes three major contributions to the literature. The first is the proposal of a hybrid feature selection approach to identify the optimal features for detecting social bots using profile features (published article [105]). The second is the proposal of the stream-based evolving bot detection framework (SEBD) to detect social bots during the evolution of a hashtag/keyword on Twitter using feature- and graph-based models. The third is the proposal of a transfer learning (TL) model (Bot-MGAT) that is used as a pretrained model for SEBD [106] with the application of probably approximately correct (PAC) learning to maintain accuracy and confidence.

The existing social bot detection tools/frameworks in the literature are summarized in Table 1.3. We focused on real-time bot detection and targeted hashtag evolution, as we believe that detection during the incident is more effective and beneficial in mitigating the damage to trends by bot accounts.

Table 1.2: Strengths and limitations of social bot detection approaches

Detection Approach	Strengths	Limitations
Feature-based	- Good overall performance, as reported in literature. - Easy to implement when trained data are available, especially because some studies provide their Python packages for testing. - Commonly used, which offers the advantage of combining various techniques, and avoid pitfalls and mistakes reported in previous studies.	- Bias toward trained dataset (includes representative samples), performance reduces with the addition of new data. - Available datasets do not include the full list of features owing to user privacy issues; therefore, not all features are available for testing and analysis. - Some techniques are based on assumptions that lack wide-scale applications. - Using the accuracy rate is not sufficient for evaluating the performance of the proposed model.
Graph-based	- Helpful for identifying influence ranking. - Uses mapping visualization to easily identify influence/rank. - Understanding links/behavior patterns (GNN). - Detect communities based on similarities/dissimilarities.	- Computational cost is high as social network size is generally large. - Simulations of network behavior can be sensitive to selected scalability measurements.
Crowdsourcing	- Humans are more capable of identifying distinct differences in the Turing test. - Suitable for small data sizes. - Useful for building a ground truth baseline.	- Requires an expert. - Lack of automation tools for simplifying the tasks. - Inconsistent quality owing to the presence of Cyborg accounts. - Incurs excessive workload and time for large data.

Table 1.3: SEBD and existing frameworks/tools for social bot detection

	Approaches		Target	Learning	Comments
	Feature-based	**Graph-based**			
Real-time spam detection (2022) [50] Accuracy 98.74%	✓		Spam or Human.	Online	- Classify a tweet to identify if it is spam or human. - Using multiclassification (deep learning) for sentiment analysis and binary classification for spam detection.
Botometer (2017) AUC: 95% Then 85%	✓		Human or Bot.	Offline	- Requires regular sample updates to be effective. - Accuracy drops when tested using new data.
Debot (2016) AUC: 94%	✓		Detect similar profiles.	Online	- The tool is not working properly anymore owing to the new regulations of Twitter that restrict API key generation. - Focus on similar profile behavior to detect social bots.
BotWalk (2017) Precision 90%	✓	✓	Identify bot followers of users.	Online	- Adaptive exploration. - Requires a bot account as a seed to search followers and verify.
BotSlayer [107] (2019) AUC: 70%	✓		Evolving misinformation.	Online/offline	- Assigns bot scores using Botometer. - Targets social bots by identifying potential amplification of information.
SEBD (This study) AUC: 98% Accuracy 96.7	✓	✓	Identify bots in a trending hashtag/keyword.	Online	- Uses the GNN approach. - Focuses on hashtags/keywords to identify social bots. - Feature-based approach for evaluating predictions.

Chapter 2: Methods and Results

This section presents the methodology used in this study. Subsequently, we discuss the methods and results for each of the three research objectives.

2.1 Research Methodology

Figure 2.1 shows an overview of the research methodology used to conduct this research and build the SEBD framework. We went through five main phases and revised them, if required, to achieve the main research goal and propose a real-time social bot detection framework for social networks (SEBD). We followed these phases to make informed decisions and ensure that the research objectives were met successfully and efficiently.

The five phases, as shown in Figure 2.1, are understanding the problem, data collection and preprocessing, feature selection (Research Objective 1), model design (Research Objective 2), and model evaluation (Research Objective 3). The description of each phase includes a primary task and procedures, as well as the instruments/tools and techniques employed. After considering these important steps, we reflected on our research methodology by addressing data collection as an essential step for ML. We then determined the predictive features for identifying bots to achieve Research Objective 1. Subsequently, a model was designed. This included building and implementing the SEBD framework to achieve Research Objective 2. Finally, the proposed framework was evaluated to ensure good on-stream performance, which reflects Research Objective 3. The main phases of the research methodology are discussed in detail in the following subsections.

In phase one, it was essential to understand the strengths and limitations of the detection methods proposed in the literature, as discussed in Section 1.5, to close the gap in this research area. A few notable remarks from the literature were addressed when building our proposed framework (SEBD)[3, 108, 109]. The first is the shortage of labeled data for detecting bots; thus, an ML model that requires a large amount of labeled data was not suitable. Therefore, we considered an semi-supervised learning (SSL) model that incorporated both labeled and unlabeled data. Second, most studies

mentioned in Sections 1.5 and 1.6 were conducted in the offline mode; therefore, there is room for real-time detection. Third, real-time detection requires considering fewer features to decrease the computational costs and processing time. Fourth, randomly detecting bots is insufficient; therefore, a detection framework was implemented from where they gather–a hashtag. Finally, combining feature- and graph-based methods was suitable for the problem because the best features of both models could be integrated.

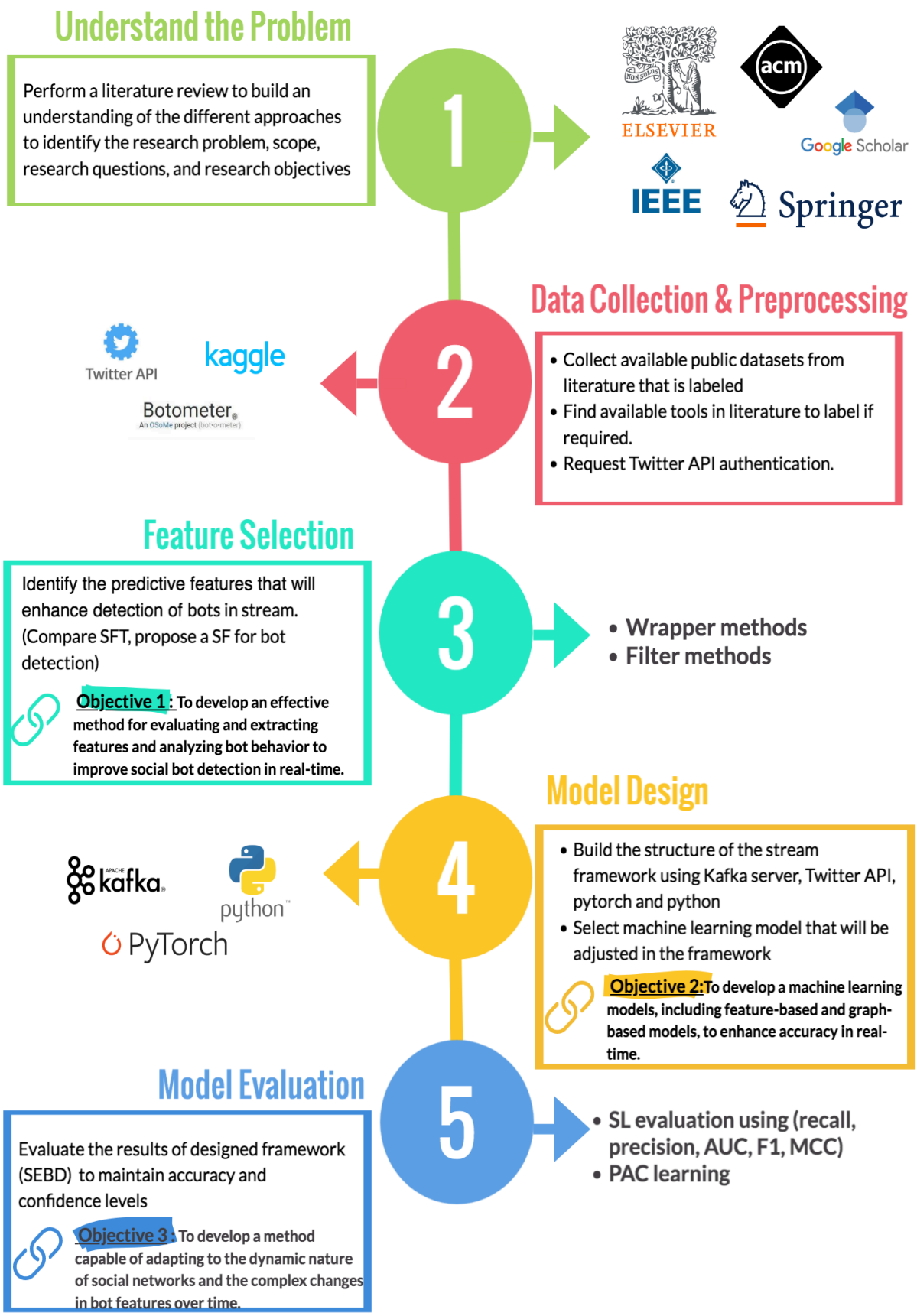


Figure 2.1: Overview of the research methodology

2.2 Data Collection and Preprocessing

ML relies heavily on the collection of relevant data. The effectiveness of an ML model highly depends on its training data. Therefore, collecting relevant and diverse data is essential for building accurate and dependable models. Hence, data quality is one of the most important factors. Building a reliable predictive model without high-quality input data that adequately represent the target population is impossible. A large dataset allows an ML model to learn more patterns and make more precise predictions. Overfitting, which indicates that the model has learned the training data so well that generalizing it to new data challenging, can be avoided by using diverse data. Thus, obtaining labeled data is integral to train and experiment with our model. The datasets used and their statistics are summarized in Table 2.1. In our experiments for building and implementing the framework, we trained our model using Twibot-20 [55] which is a recent diverse benchmark dataset containing a small proportion of labeled samples. It includes both profile features and graph links of follower/following relationships. Most publicly available datasets focus only on profile features or tweet content, and only a few constitute graph links.

Moreover, we used a labeled dataset [110] available on Kaggle.com, which was also used in [63], to build a hybrid model for identifying predictive features. This dataset is a combination of previous datasets (from 2016–2019). The author of this dataset removed suspended accounts that could not be retrieved from the Twitter API after Twitter suspended them. Therefore, this dataset was imbalanced. It was used in several studies mentioned in [41] and is available separately in [111].

The last two datasets listed in Table 2.1 were downloaded from NodeXL, and both were constructed from hashtags. Both datasets were used as testing and validation sets in our published work [106] as they have profile features and interaction links for building the graph.

As part of the data collection for our model, we required the Twitter API to retrieve stream and process data, as shown in Table 2.2. These hashtags were collected and processed using SEBD. Group 1 hashtags were collected on September 2022, while Group 2 hashtags were collected on December 2022. In addition, we used a baseline tool

proposed by Indiana University, called Botometer, which employs SL [112] and has been used in the literature to label unlabeled data. We used it as a baseline tool for the evaluation. For the ground truth dataset, we used a dataset available on GitHub,[1] as listed in Table 2.1. The author of this dataset used Botometer V4 to label accounts obtained using the Twitter API [113]. The dataset comprises more than 100K samples with 33 profile features, and both classes are balanced. We refer to them as feature profile metadata.

Table 2.1: Statistics of the used datasets

Dataset	Released	Nodes	Edges	Features	Labeled	Balanced	Source
Kaggle.com	July 2020	-	-	P	37,438	No	[63]
Github	July 2022	-	-	P	105 K	Yes	[113]
Twibot20	May 2021	229,573	455,958	P- T- N	11,826	Yes	[55]
Climate hashtag	April 2022	5,664	15,651	P- T- N	-	-	[114]
Russia hashtag	May 2022	4,941	6,944	P- T- N	-	-	[115]

P: Profile T: Tweet N: Network

Table 2.2: Datasets that were retrieved and processed by SEBD

Group	Hashtag	Domain	Total number	Trending	Combined	Worldwide
1	#WorldCup2022 #Russia OR Ukraine #Bitcoin #iOS16 #NotMyKing #gamescom2022	Sport Political Financial Technology Political Entertainment	4705	✓	✓	✓
2	#AEWDynamite	Sport	2,355	✓	✓	✓
	#AbbottElementary			✓	✓	
	#Mozz			✓	✓	
	#Cowboys	Sport	2,035	✓	✓	
	#PMQs	Political	3,070	✓	✗	✓
	#Oscars2023	Entertainment		✓	✓	
	#ClimateCrisis	-	1,540	✓	✓	✓
	#iOS16	Technology	1,986	✗	✗	
	#ChatGPT	Technology	1,681	✗	✗	
	#TheLastofus	Entertainment	1,972	✓	✗	✓
Total			19,344			

The preprocessing step is important to prepare the training data for the classifier. First, we removed all duplicates and any highly inconsistent features, such as *description*

[1]https://github.com/Niyaz94/Bot-Detction-On-Twitter/tree/main/Datasets

[62], from the training data. We replaced the account type values with zeros and ones for *humans* and *bots*, respectively. Furthermore, we ignored *location* and *description* features because they contained many values, and we removed any features with null values, such as *is_translator*. For categorical features, we used Boolean values of 1 for true and 0 for false to accelerate the processing. In conclusion, we applied standardization to all numerical features to obtain a mean of 0 and a standard deviation of 1.

2.3 Model Evaluation

To evaluate the predictions of SEBD, we used the following commonly used evaluation metrics: accuracy, F1 score, and Matthews correlation coefficient (MCC). Accuracy was used to evaluate the model correctness. Recall indicates the completeness of the model and precision indicates its exactness or predictive power. The F1 score and MCC were used as balanced evaluation metrics for the binary classification tasks. The metrics are defined as follows [116]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = TP/(TP + FP) \quad (2)$$

$$Recall = TP/(TP + FN) \quad (3)$$

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

where TP, TN, FN, and FP denote true positive, true negative, false negative, and false positive samples, respectively.

We used the holdout method (split 30%) with cross-validation to ensure that there were no overfitting issues in our model. In addition, we compared the prediction results of our model with those of some baselines. We also compared its performance with those of other ML classifiers in terms of F1 score, precision, and recall. Moreover, we used

baseline tools, such as Botometer, and benchmark datasets to evaluate the performance of the SEBD framework.

It is good to mention that we did not include concept drift or concept shift to the evaluation due to the limitation of Twitter API. The rate limit and pause during retrieving data from Twitter has lead to offline evaluation.

2.4 Experimental Environment

All experiments were performed on a workstation comprising an Intel Core i7-1087 CPU 2.4 - 5.3GHz, 32 GB RAM, and an NVIDIA GeForce RTX 3070 Max-Q GPU with 6 GB of VRAM. In addition, all ML models were built using the Python scikit-learn library. All graph-learning models were implemented using the PyTorch geometric library (PyG)[2].

2.5 Objective 1: To Develop an Effective Method for Evaluating and Extracting Features and Analyzing Bot Behavior to Improve Social Bot Detection in Real-time

Feature selection is an essential step in the ML pipeline, particularly for detection models. Essentially, it entails selecting a subset of useful features from a wider set of potential features to improve the accuracy and efficiency of an ML algorithm. Therefore, feature selection can significantly affect the performance of bot detection models because irrelevant or redundant features can result in overfitting, degrade model interpretability, and increase computational complexity. Thus, carefully analyzing the most instructive characteristics for a specific task is crucial.

ML constitutes multiple methods for selecting features, including filters, wrappers, and embedding. Filtering techniques employ statistical metrics such as correlation or information gain to rank features according to their importance to the target variable [117]. The highest-ranked characteristics are then selected for further analyses. This method is computationally efficient and can handle high-dimensional data. However, it may not consider feature interactions or complex dependencies. In contrast, wrapper approaches integrate feature selection into the model selection process by assessing various subsets of features and selecting the best-performing subset. This

[2] https://pytorch-geometric.readthedocs.io/en/latest/index.html (accessed on 1 September 2021)

method is computationally more expensive but considers feature interactions and dependencies.

Twitter features are categorized into profiles, tweet networks, and temporal features, as shown in Figure 2.2. A previous study found that approximately 1,000 features can be obtained from a Twitter account [5]. Several characteristics can be used to differentiate human users from bots. For example, account age can be used to identify bot accounts, as they are typically created recently, whereas accounts of real users are generally older. The follower-to-following ratio can be used to distinguish bots because they have a higher follower-to-following ratio than human users. However, it was discovered that this feature was changed in newer bot versions [118]. The first generation of bots generated a significant number of tweets in a short amount of time, whereas human users tweeted less frequently on average. However, Twitter's automation policy has limited this behavior, as any account with massive automated activity is automatically suspended [119]. Therefore, bots have adapted by reducing their excessive generation behavior to avoid detection. Additionally, bots may exhibit similar network patterns, such as a large number of connections with other bots. However, they may exhibit activity patterns that differ from those of human users, such as tweets during odd hours or participating in repetitive activities [120]. Their outputs may exhibit a constant sentiment, but human users are more likely to express a variety of feelings. Another feature that can distinguish bots is the generation of similar or identical content, because human users are more likely to generate unique content.

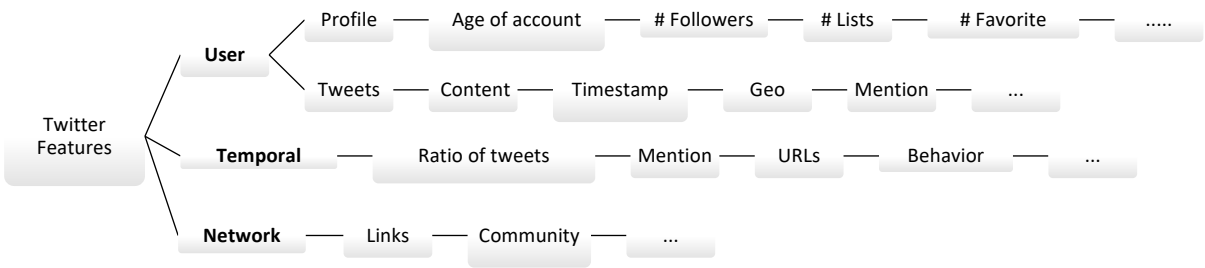


Figure 2.2: Categories of Twitter features

2.5.1 Objective 1: Method

We have proposed a hybrid method [105] to identify the optimal features for detecting social bots using only profile features. We used the kaggle.com dataset for our experiments, as stated in Table 2.1. The algorithm for finding predictive features integrates wrapper and filter methods to set the ceiling number (k) for the optimum size of the feature subset, as shown in Figure 2.3. Subsequently, SL was applied to compare the performances of the RF, neural network (NN), NB, and SVM classifiers for different feature subsets. We adopted these classifiers owing to their good performances in previous studies [3]. The best feature subset selection performance was reported for each classifier. This contributes to the Research Objective 1 by answering the first research question. We used cross-validation to train our data on all the features. The results showed that the learning model with RF achieved an accuracy of 79.2%. However, the model using all features was unable to identify more than 50% of bots' classes with a recall score of 37%, and able to identifying nearly all human accounts with a recall of 99.8 % and average AUC of 93.3%. However, the purpose was to determine the optimal features that could increase the performance of our learning model for identifying bot classes without affecting the human classes. In particular, we handled imbalanced data with a bot-to-human ratio of 33:67.

We investigated four feature selection techniques, as listed in Table 2.3. We studied the Twitter profile features to enhance bot detection. The proposed method offers a hybrid technique for identifying the predictive features in Twitter profile metadata. We used the filter and wrapper techniques to determine the appropriate feature subset. The performance results for different subsets of features obtained using the algorithm shown in Figure 2.3 are presented in Table 2.3.

Algorithm 1: *Finding the optimal predictive features*

Input: Dataset *D* as a matrix of (N × M), where N is the number of features F=$\{f_1, f_2, \dots, f_N\}$, and M is the number of Twitter accounts A= $\{a_1, a_2, \dots, a_M\}$, *FSM ={m: m is a* filter feature selection method}, and CA = {c: c is a classifier algorithm}

Variables ***k:*** to save total number of features,
b*:** to save *k* features, b_{Temp} : to save current ***k features
PTC: to save Precision of Target Class, **RCT**: to save Recall of Target Class, **AUC**: to save Area Under the Curve. $\mathbf{PTC_t}$: to save current Precision of Target Class, $\mathbf{RTC_t}$: to save current Recall of Target Class, $\mathbf{AUC_t}$: to save current Area Under the Curve

Output: Finding the best features (b) for each m in FSM

```
1:  Run wrapper subset evaluation method
2:  Set k = total number of features in the subset generated by the wrapper method
3:  for m in FSM
4:      Run m on F to get all features ranked and sorted in descending order
5:      b = {f_i: i = 1, ..., k}
6:    for c in CA
7:        Run c on b to get PTC, RTC, AUC
8:        do
9:            k = k − 1;     // removing the least weight feature
10:           b_Temp = {f_i: i = 1, ..., k}
11:           Run c on b_Temp to get PTC_t, RTC_t, AUC_t
12:           If (PTC_t > PTC      and
13:               RTC_t > RTC      and
14:                AUC_t > AUC )   then
15:                PTC ← PTC_t
16:               RTC ← RTC_t
17:               AUC ← AUC_t
18:               b = b_Temp
19:           else if ( (PTC − PTC_t)/PTC < 5%   and
20:                     (RTC − RTC_t)/RTC < 5%   and
21:                     (AUC − AUC_t)/AUC < 5%)  then
22:                         if AUC_t > AUC then
23:                            PTC ← PTC_t
24:                           RTC ← RTC_t
25:                          AUC ← AUC_t
26:                          b = b_Temp
27:           else  Break
28:       while ( k > 1 )
29:       Output b;
30:   end for     // classifiers
31: end for       // features selection method
```

Figure 2.3: Algorithm for finding predictive features

2.5.2 Objective 1: Results

For feature subset selection, the cross-validation evaluation attribute (CVEA) approach with six features performed the best, as shown in Figure 2.4. In general, the RF classifier performs better than the NN classifier for identifying bot accounts. Neither the NB nor the SVM performed well in the experiments. RF achieved an accuracy of 89%, AUC of 94.3%, precision of 83.8, and recall of 83.3% for both classes. The RF with

CVEA achieved the best results with only six profile features. The precision/recall curve is shown in Figure 2.5.

For more details on the proposed method, see Publication I in Publications. We focused on profile features because the aim was to minimize the computational cost of in-stream feature processing. Additionally, several studies [51, 59, 61] have shown that profile features are predictive features for bot detection problems. A recent study found that six profile features were predictive for bot detection [121]. In fact, profile features are predictive enough, and adding tweet features increases accuracy by only 4% [59]. However, the selected features must be regularly reevaluated because bots tend to adapt to avoid detection or comply with new Twitter rules. Therefore, in the SEBD framework, we use more features than those in our previous model [105].

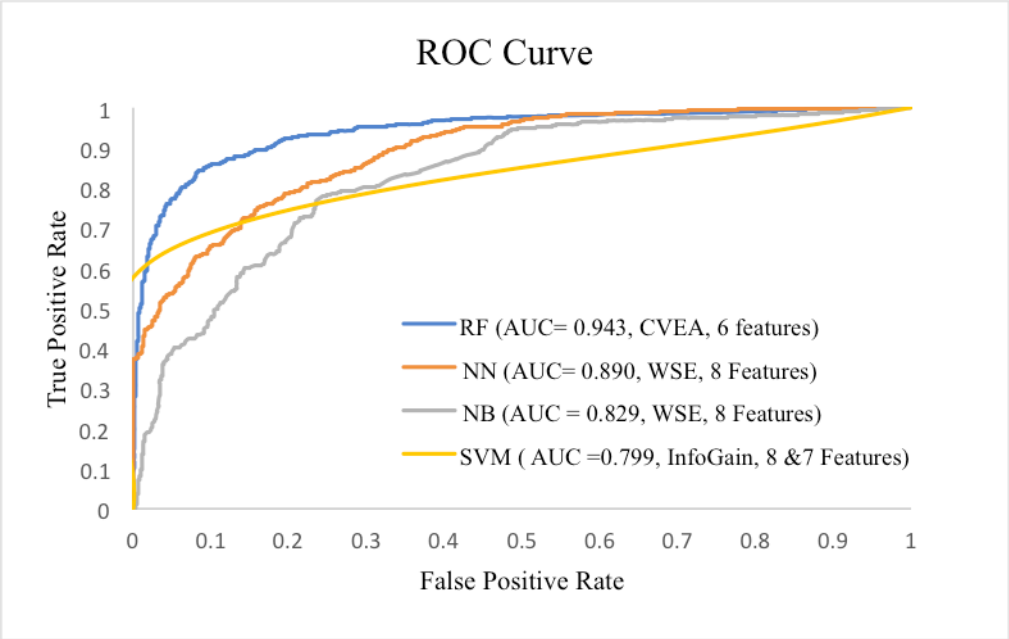


Figure 2.4: Results of the proposed feature selection methods

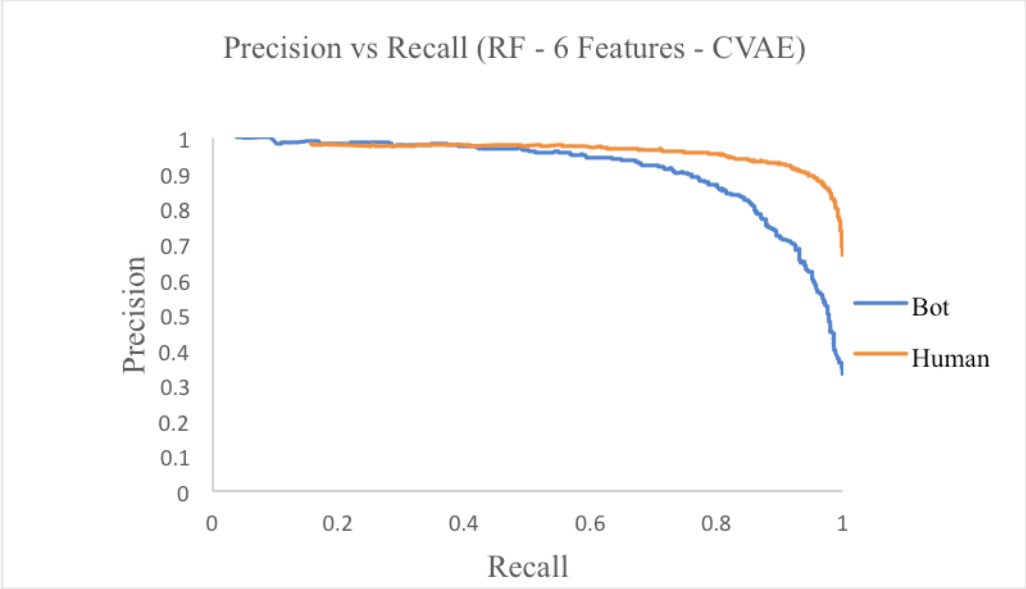


Figure 2.5: Precision vs Recall curve for CVEA method

Table 2.3: Performance for feature subset selection for the bot class

Feature Selection Method	Classifier	# Of Features	Bot P	Bot R	AUC	Bot PRC
Correlation Attributes	RF	8	0.801	0.77	0.898	0.822
		7	0.8	0.768	0.9	0.822
		6	**0.807**	**0.768**	**0.903**	0.822
		5	0.682	0.694	0.824	0.691
	NB	8	0.591	0.636	0.769	0.572
		7	0.59	0.636	0.769	0.569
		6	**0.606**	**0.652**	**0.787**	0.633
		5	0.595	0.633	0.776	0.621
		4	0.589	0.604	0.78	0.628
		3	0.757	0.266	0.766	0.589
	SVM	8	1	0.375	0.688	0.583
		7	1	0.375	0.688	0.583
		6	**1**	**0.39**	**0.695**	0.592
		5	0.662	0.378	0.641	0.457
	NN	8	**0.658**	0.581	0.826	0.679
		7	**0.658**	0.557	**0.827**	**0.688**
		6	0.631	**0.652**	0.825	0.672
		5	0.65	0.514	0.792	0.64
Information- Gain	RF	8	1	0.37	0.908	0.854
		7	1	0.37	**0.913**	**0.862**
		6	1	0.37	0.902	0.846
		5	1	0.37	0.866	0.791
		4	1	0.37	0.812	0.724
	NB	8	**0.688**	0.277	0.798	0.586
		7	0.65	0.551	**0.812**	0.637
	SVM	8	**0.872**	**0.646**	**0.799**	0.68
		7	**0.872**	**0.646**	**0.799**	0.68
		6	0.77	0.638	0.772	0.612
	NN	8	0.952	0.383	0.861	0.791
		7	0.949	**0.388**	**0.865**	**0.799**
		6	0.941	0.383	0.858	0.789
		5	**0.956**	0.383	0.842	0.774
		4	0.956	0.385	0.845	0.776
		3	0.948	0.385	0.837	0.771
		2	1	0.37	0.805	0.729
CV Attribute Evaluation	RF	8	0.873	0.644	0.923	0.876
		7	**0.882**	0.649	0.93	0.885
		6	0.838	**0.833**	**0.943**	0.91
		5	0.825	0.812	0.94	0.899
		4	0.841	0.812	0.938	0.9
		3	0.844	0.802	0.932	0.892
		2	0.674	0.676	0.822	0.68
	NB	8	**0.577**	**0.127**	**0.794**	0.564
		7	0.504	0.093	0.753	0.517
	SVM	8	1	0.395	0.697	0.595
		7	1	0.395	0.697	0.595
		6	1	0.367	0.684	0.577
	NN	8	**0.708**	0.733	0.852	0.726
		7	0.652	0.741	**0.854**	**0.717**
Wrapper Subset Evaluation	RF	8	0.992	0.377	0.914	0.865
	NB		0.684	**0.536**	**0.829**	0.485
	SVM		0.648	0.676	0.747	0.546
	NN		0.879	0.514	**0.89**	**0.831**

2.6 Objective 2: To Develop a Machine Learning Models, Including Feature-Based and Graph-Based Models, to Enhance Accuracy in Real-time

The main goal of this study was to propose a real-time bot detection framework that overcomes the limitations of existing models. The model design phase began after the completion of the first phase: understanding the problem, data collection and preprocessing, and selecting features. Each of these processes is essential for developing a suitable model that can address the research questions of this study. After completing the first three steps, we had to select an ML model that could help us achieve Research Objective 2. As mentioned previously, we required an SSL model because both labeled and unlabeled data were involved. This saved the effort required for labeling, works well with the model in a straightforward manner by using labeled data to predict unlabeled data points.

The issue of labeling can be resolved by using GNN models [103]. The SEBD model was built using a multi-view graph attention network (GAT) model, as it can learn the complex interactions between nodes and capture the structural information of the graph, which is essential for bot detection. The following subsections elucidate the methods and results for Research Objective 2 reflecting in the SEBD structure and our pretrained model (bot multi-view graph attention network (Bot-MGAT)). We have three published works on this topic: the stream framework structure was published in [122], the Bot-MGAT model used in the SEBD was published in [106] and the SEBD framework in [123].

2.6.1 Objective 2: Method

As shown in Figure 2.6, the SEBD framework comprises three distinct phases. The first is data collection, which begins by submitting a hashtag to the Twitter API to obtain recent and popular tweets related to a given hashtag or keyword. The API retrieves the necessary information about user accounts gathered from tweets. From these tweets, we gathered several profile features.

The second phase is streaming. To build the SEBD structure, we used the Apache Kafka server [124], as described in Section 2.5. Apache Kafka is a distributed streaming platform used to build real-time streaming applications and data pipelines. Kafka

comprises four main components: brokers, topics, producers, and consumers. Its primary component is the Kafka server, which offers a messaging infrastructure that enables producers to post messages under topics and consumers to consume those messages within the topics.

Brokers are Kafka host nodes tasked with storing and processing messages. These are the identified feeds or categories from which producers can send messages and consumers can read them. The topics can be divided into several categories to facilitate parallel processing and scalability. Producers are application APIs that send messages to Kafka brokers. Consumers indicate the applications that read the messages sent by Kafka brokers. Consumers can be categorized into groups based on the distinct subset of topics they read. The Kafka Connect tool enables synchronization between Kafka and other systems by providing connections for common data sources and sinks. The Kafka Stream library is used for building real-time stream processing applications on top of Kafka. Additionally, we used the ZooKeeper server [125] to store the results of our model.

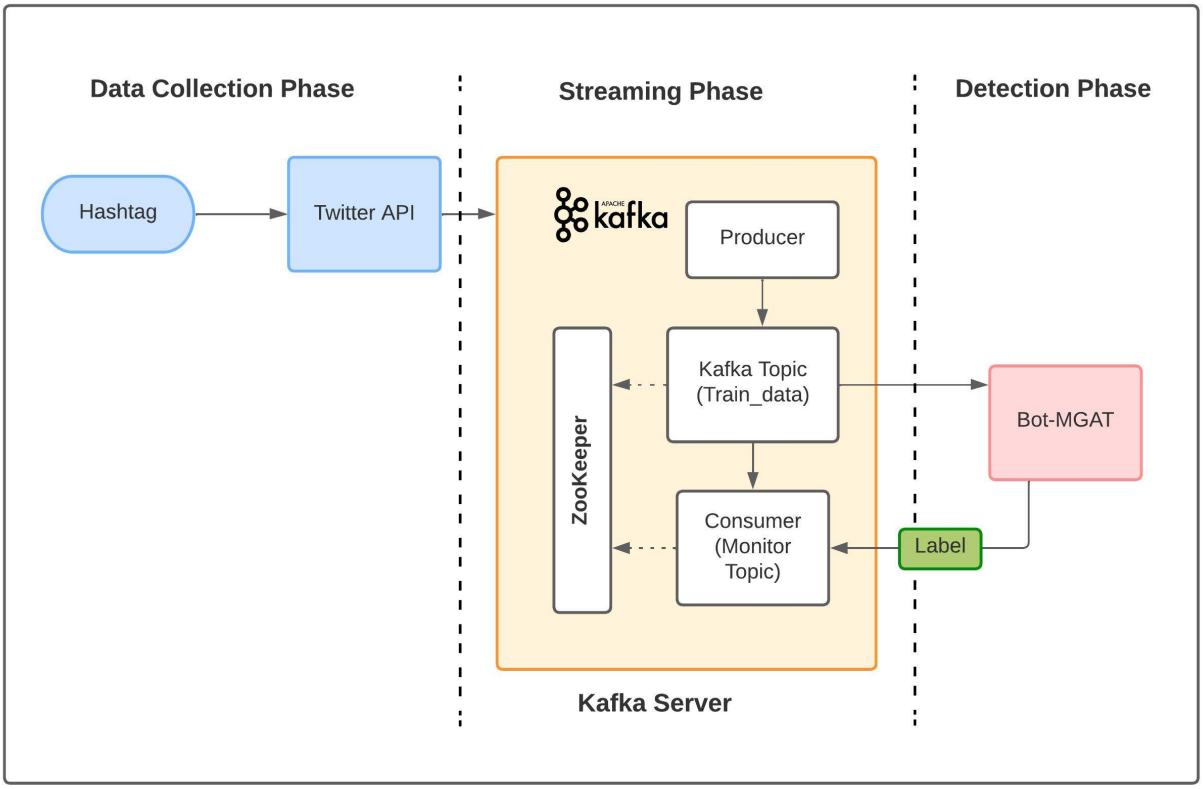


Figure 2.6: SEBD framework

In the final phase, detection, we predicted the label of each account using out pretrained model (Bot-MGAT), as shown in Figure 2.7. Bot-MGAT [106], built using GATs, is the core of SEBD, as it is the ML model that classifies accounts into humans or bots. GATs are graph neural networks wherein nodes on the graph's edges are assigned different weights to indicate their relative importance. Weighting is performed through self-attention, which combines the embeddings of nodes and their neighbors and performs a linear transformation [126]. In a multi-view graph, which is a type of relational graph, the same collection of nodes can be observed from numerous angles (or, technically, multiple graphs with the same nodes but different edges). Bot-MGAT performs relationally oriented self-attention, which means that the attention method is applied to each graph view, and the resultant node embeddings are concatenated to perform classification tasks [106]. We trained Bot-MGAT using the benchmark dataset Twi-Bot20 [55]. The results were saved in a csv file to perform an offline evaluation using the RF classifier. For more details, please refer to our previous study [106]. The entire SEBD flowchart is shown in Figure 2.8. Screenshots of SEBD are provided in Appendix.

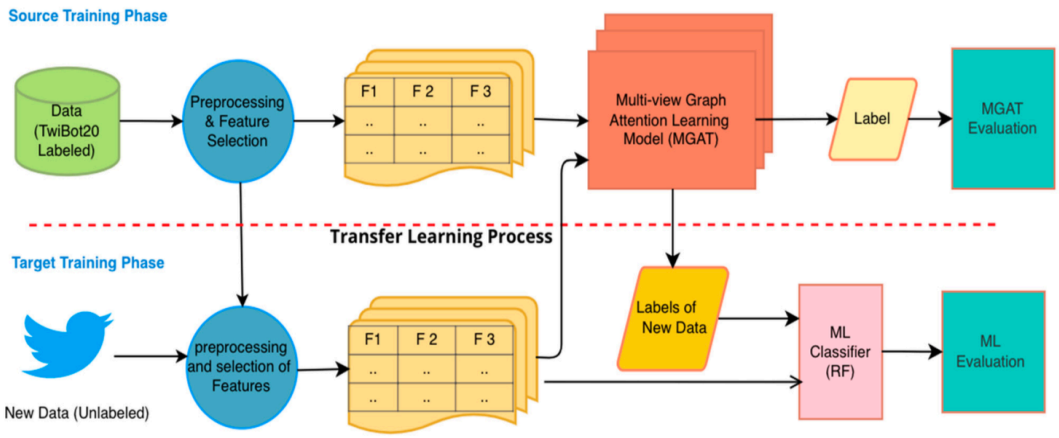


Figure 2.7: Bot-MGAT Model with Transfer learning

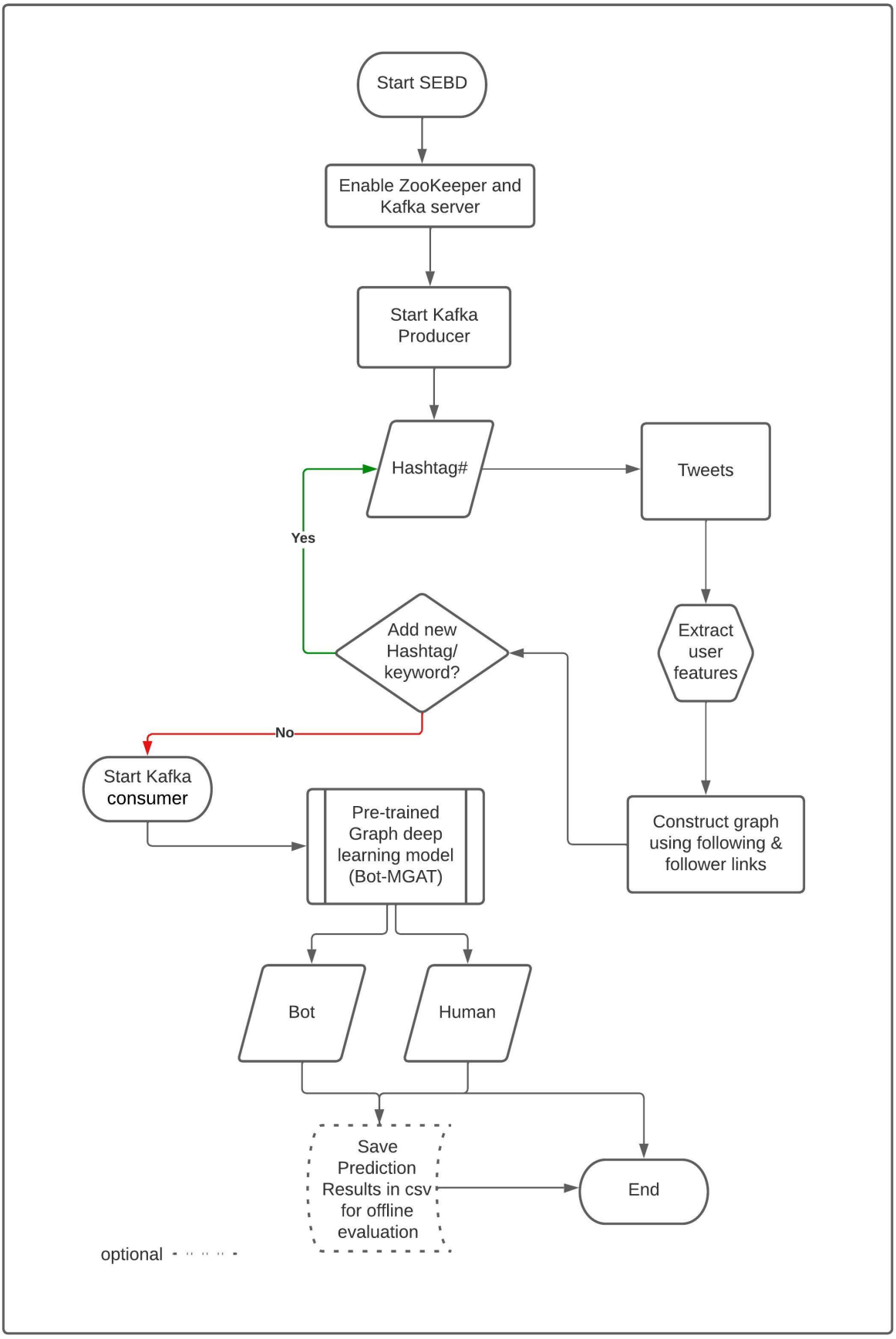


Figure 2.8: SEBD flowchart

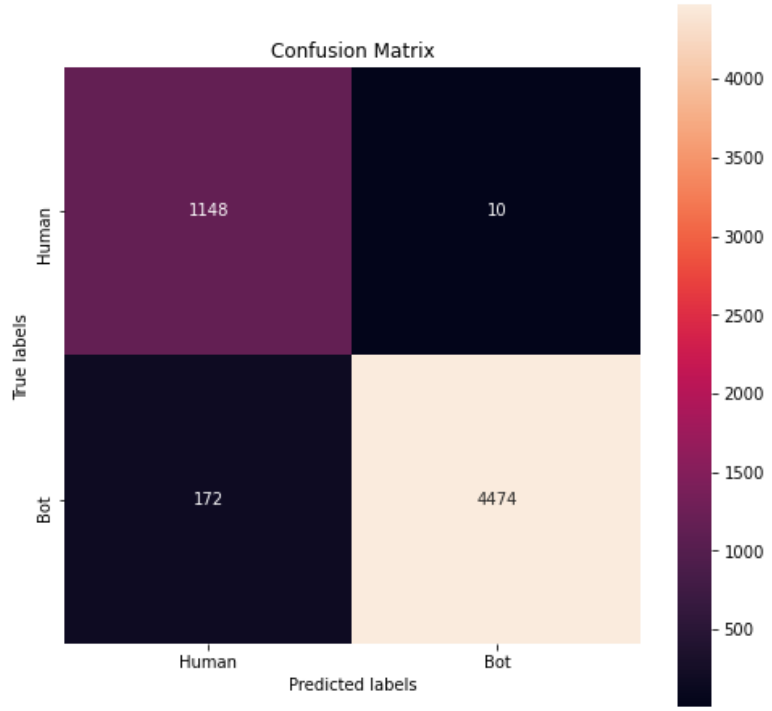
To address the issue of overfitting in early phases, we considered reducing the complexity of Bot-MGAT. We decreased the number of parameters using regularization techniques, such as L1 (lasso regularization) or L2 (ridge regularization), and applied early stopping during training to prevent the model from continuing to learn on the training data at the expense of generalizability. Early stopping is a technique that lets you provide an arbitrary large number of training epochs and terminate training as soon as the model's performance on the validation dataset stops advancing [127].

2.6.2 Objective 2: Result

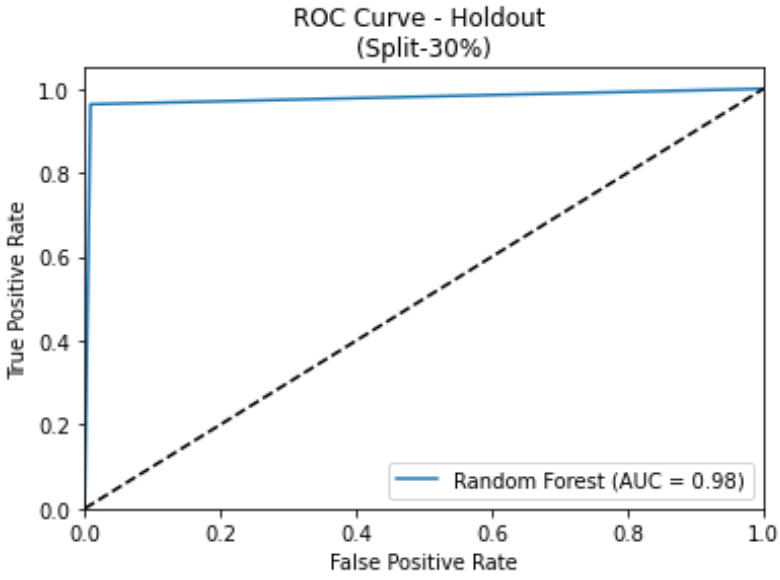
We compared the performance of the SEBD framework with those of four evaluation methods. The first was the holdout method with cross-validation, wherein 70% data was used for training and 30% for testing to evaluate the predictions for the datasets listed in Table 2.2. SEBD achieved an accuracy of 96.9% with an F1 score of 98%, recall of 96.2%, precision of 99.8%, and MCC of 91%. These results indicate that the SEBD can effectively identify bot accounts. As shown in Figure 2.9, these results were supported by the confusion matrix and AUC. Table 2.4 presents the results of the SEBD framework and some baselines.

We further evaluated the SEBD framework using a baseline tool to label the data for which the classes were predicted by our model. We used Botometer [112] because it is widely used in the literature and adopts automation scores to identify the likelihood of an account being a bot or human using a threshold value. This metric is called the complete automation probability (CAP) [112]. If the CAP threshold is 0.7, any account with CAP > 0.7 is likely to be a bot; otherwise, it is a human. We used two different thresholds, 0.5 and 0.75, as listed in Table 2.5. We have chosen 0.5 as a middle threshold and 0.75 as indicated by literature [128] to compare both performances. The ROC curve results are shown in Figure 2.10 and Figure 2.11, which indicate that the threshold of 0.5 performed better than 0.75 in terms of F1 score, recall, and precision. This indicates that bot accounts tend to reduce and adjust their automation behavior to avoid being detected. Threshold 0.75 showed a drop in accuracy, F1, and precision, but an improved recall score. A recall score of 86% indicates that the model correctly identified 86% of the

positive instances in the dataset. A precision score of 68% indicates that out of all the instances classified as positive, only 68% were actually positive. In other words, 32% of human accounts were labeled as bots by Botometer.



(A)



(B)

Figure 2.9: SEBD perfromance results (A) SEBD confusion matrix (B) SEBD results for the holdout method and cross-validation with RF.

Table 2.4: Results of SEBD and other baselines

	Approach	Accuracy	F1	MCC
BotRGCN [59]	RGCN	0.8462	0.8707	0.7021
Alhosseini et al. [71]	GCNN	0.94	0.84	-
SEBD (our)	MGAT	0.969	0.98	0.91

Table 2.5: Botometer results using group 1 dataset

CAP Threshold	Accuracy	F1	Recall	Precision	MCC
0.50	0.7698	0.8645	0.8668	0.8623	0.1002
0.75	0.6491	0.7665	0.8843	0.6765	0.1254

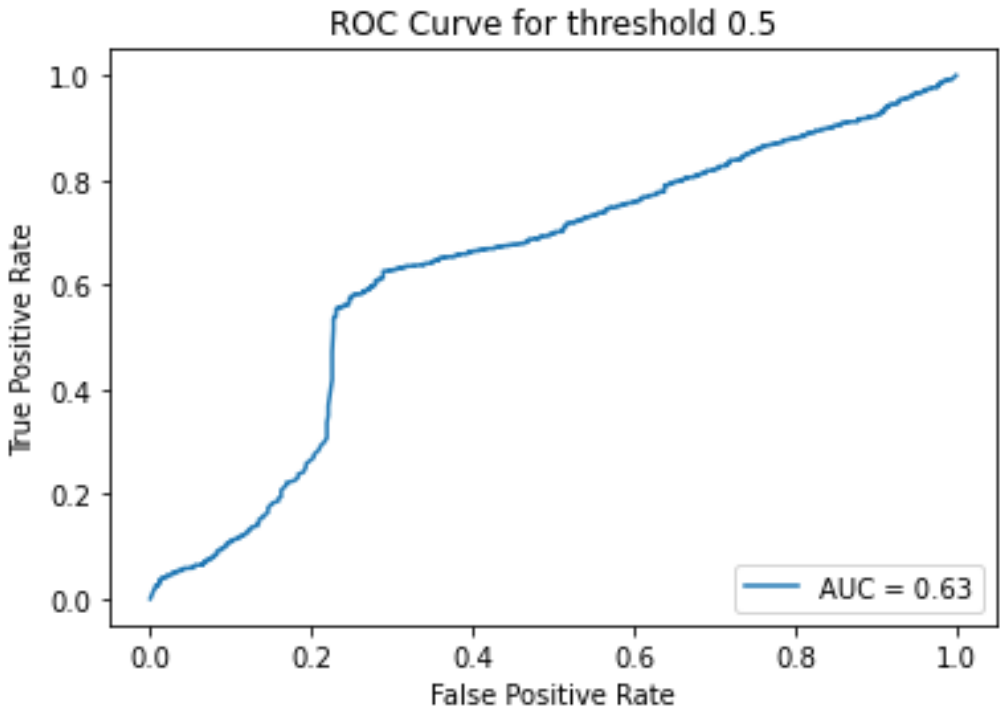


Figure 2.10: ROC curve using Botometer with CAP threshold 0.5

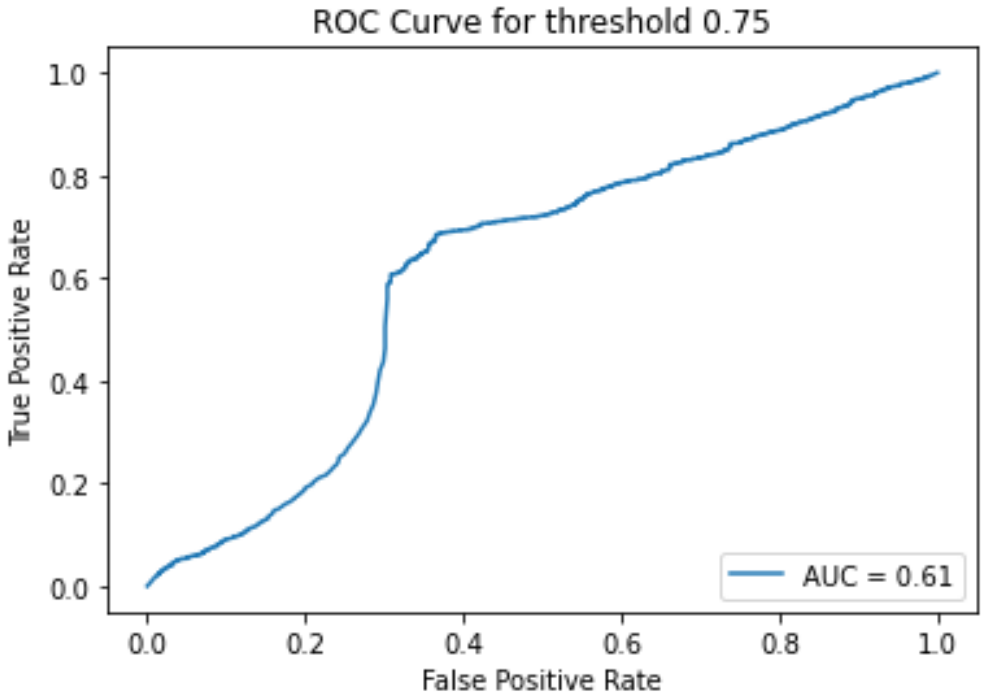


Figure 2.11: ROC curve using Botometer with CAP threshold 0.75

Furthermore, we evaluated SEBD using a benchmark dataset known as the GitHub dataset, as listed in Table 2.1. This dataset had not been previously used for this purpose. We trained a random forest with cross-validation using the GitHub dataset, which was labeled using the Botometer tool. The dataset consisted of more than 100K samples. We then employed the train/test method to validate the dataset that SEBD predicted. The results are shown in Table 2.6, and the AUC curve is depicted in Figure 2.12.

Table 2.6: Result of benchmark dataset with RF train/test using group 1 dataset

Dataset	Accuracy	F1	Recall	Precision	MCC
Ref [113]	0.8273	0.8967	0.8708	0.9245	0.3780

The results indicate that the SEBD performed well and achieved precision, recall, and F1 scores of 0.93, 0.87, and 0.90, respectively. This indicates that the SEBD succeeded in identifying bots and classifying the samples into humans and bots.

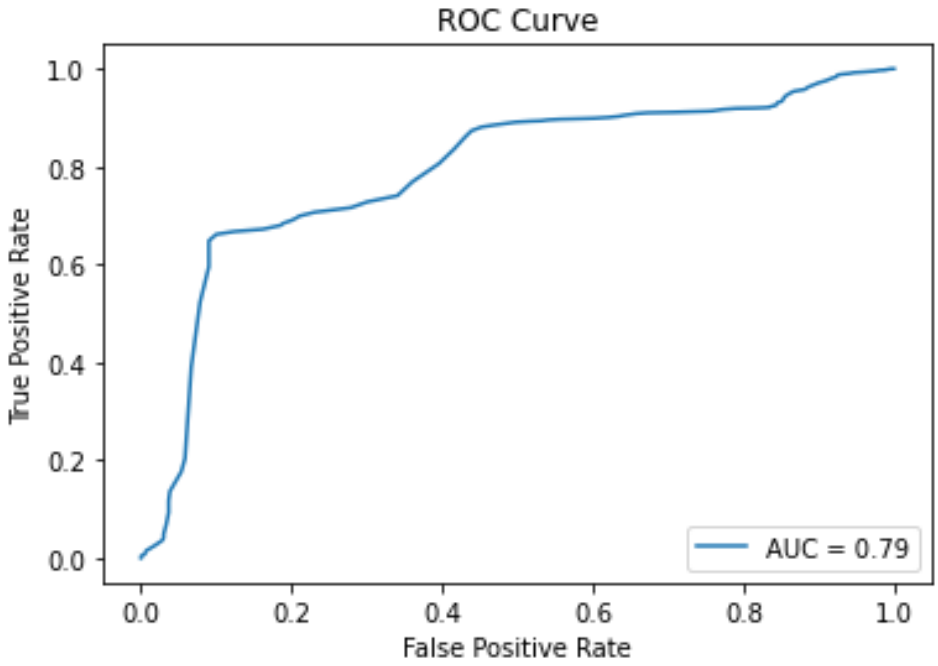


Figure 2.12: ROC curve for benchmark dataset using train/test with group 1 dataset

The last evaluation was conducted using multiple ML algorithms with cross-validation (10 folds) for SEBD and all the datasets listed in Table 2.2. SVM, RF, KNN, NB, and ensemble algorithms were employed, and their F1, recall, and precision scores were compared. The results are illustrated in Figure 2.13 and 2.14. They indicate that the SEBD achieved recall scores higher than 95%; therefore, it can identify bot accounts. They also show that the model is accurate in terms of precision scores: 98.5% for RF, 96% for ensemble, and 95.9% for KNN. SVM and NB achieved precision scores of 90.6% and 90.5% respectively. RF and ensemble outperformed other classifiers in terms of AUC (RF = 95.3% and ensemble = 90.7%). Generally, RF and ensemble algorithms are effective tools for bot detection because they are robust, adaptable, and capable of processing complex and noisy data. RF models are constructed using decision trees trained on randomized subsets of data to make them resistant to noise. This implies that they are less sensitive to noise and outliers in the data than other classifiers, which may be advantageous for bot detection problems if the data are noisy or contain outliers.

RF can automatically select the most pertinent features for a given problem, which might be useful for bot detection in cases wherein a significant number of features must be examined. This reduces the risk of overfitting and enhances the generalizability

of the classifier. Ensemble approaches, such as bagging and boosting, combine numerous models to generate a final prediction. This strategy can enhance the classifier performance by reducing the prediction variance and bias. This is particularly beneficial for bot detection because it helps manage the complexity and variations in bot behaviors.

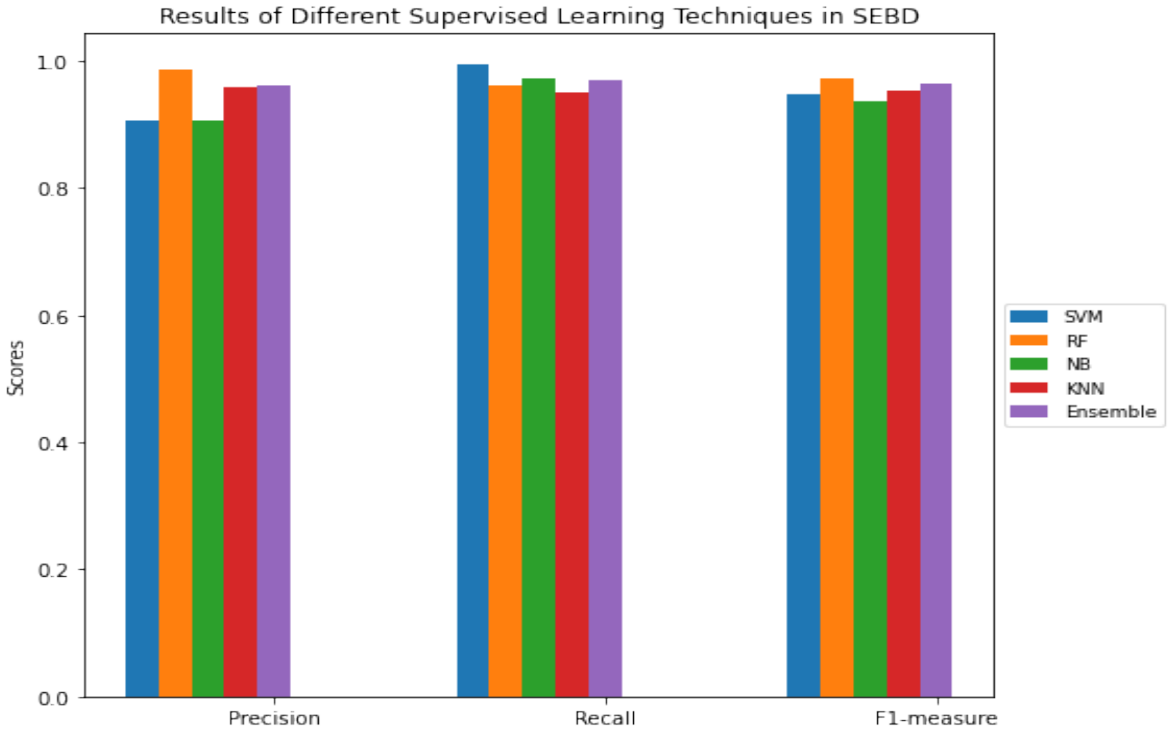


Figure 2.13: Results of SEBD with different SL classifiers

Similar results were obtained by applying different techniques to Bot-MGAT for the Twibot-20 dataset to avoid or reduce the overfitting issue. As presented in Table 2.7 applying L1 and L2 regularizations, and not applying any regularization yielded very similar results. However, the performance decreased when the multi-view in the GAT model was stopped early and removed. Figure 2.15 shows the results of Bot-MGAT without regularization, and we can see that the validation loss begins to deviate from the training loss after approximately 50 epochs. This may be a sign of overfitting, although we included dropouts in our model. The effects of L1 regularization are shown in Figure 2.16, and Figure 2.17 shows the effect of adding L2 regularization. Based on the MGAT model, we included only the weights of the GAT layers (self-attention layers) for

regularization. This aids in making the weights of the views close to each other because they might share information between them.

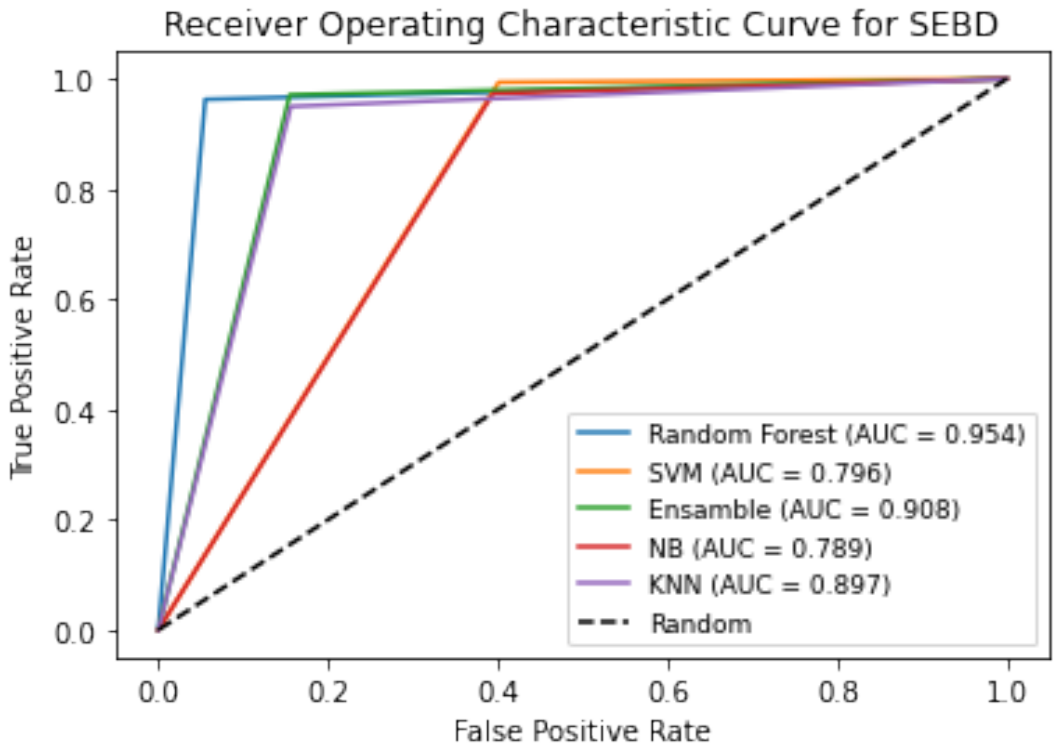


Figure 2.14: ROC curves of the SEBD framework with different SL classifiers

Table 2.7: Over-fitting techniques on Bot-MGAT

Model	ACC	F1	MCC
Without regularization	**0.8098**	**0.8485**	**0.6491**
L1	0.8047	0.8444	0.6378
L2	**0.8098**	0.8483	0.6482
Early stopping	0.7997	0.8395	0.6234
GAT without views	0.7785	0.8165	0.5636

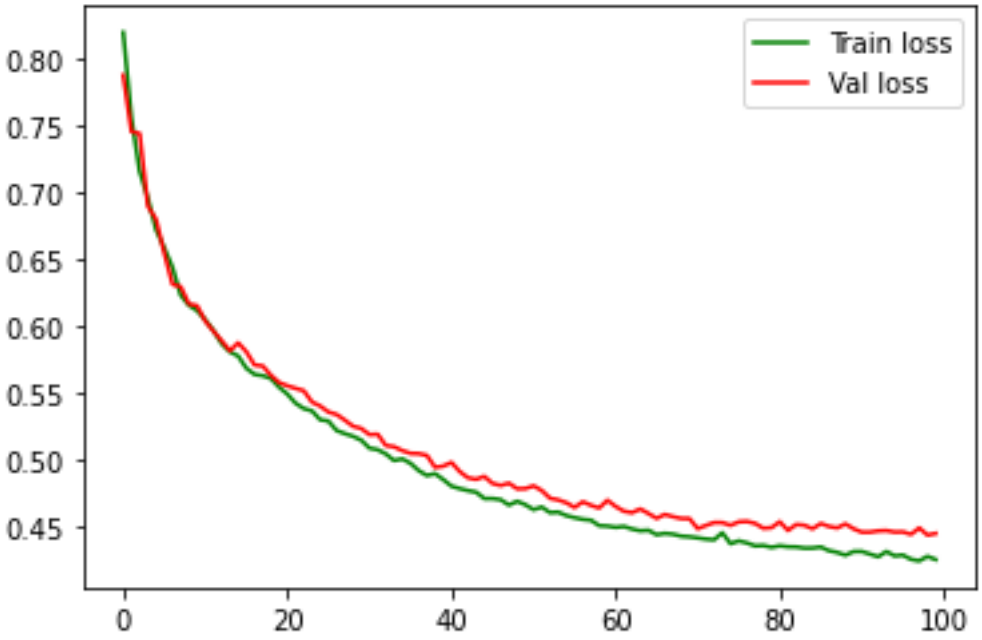


Figure 2.15: Bot-MGAT without regularization for validation loss and training loss

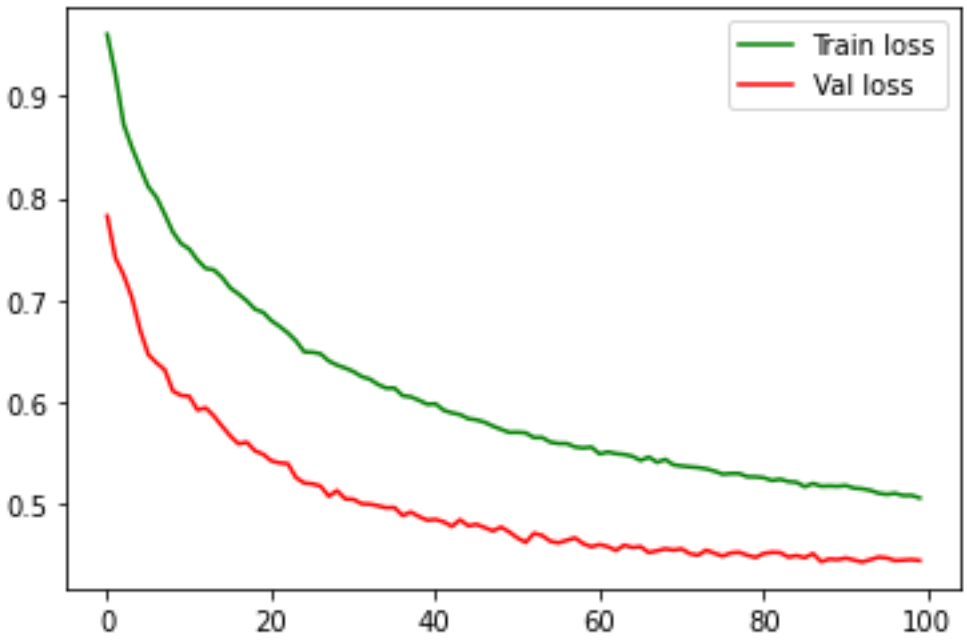


Figure 2.16: Effect of regularization L1 on Bot-MGAT

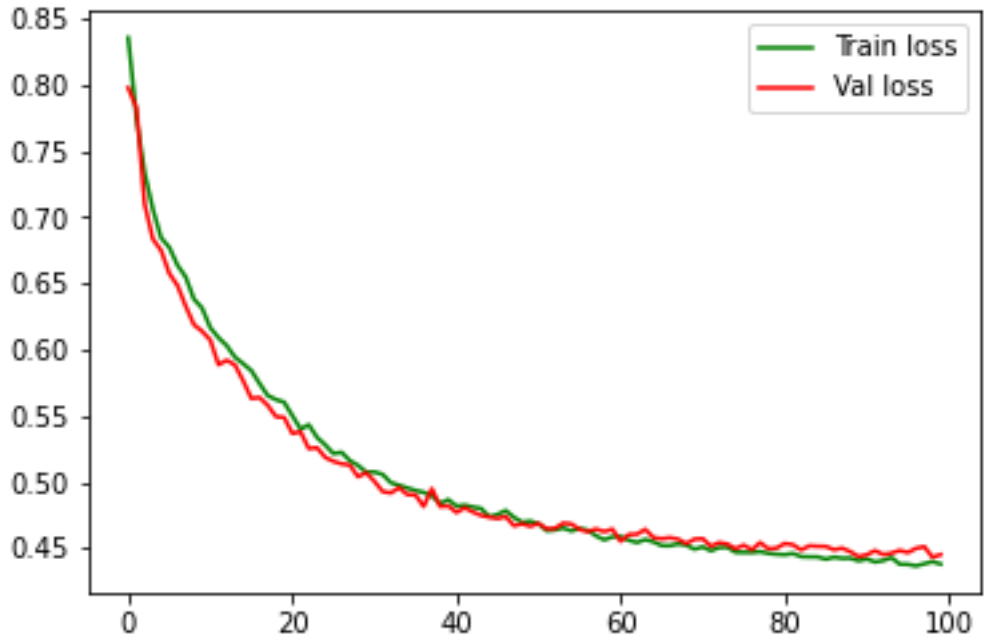


Figure 2.17: Effect of regularization L2 on Bot-MGAT

For early stopping, we must select a metric to monitor and a patience parameter to select the number of epochs to wait for before deciding to stop. The selected monitoring metric was the validation loss and the patience parameter was 5–10 epochs. The results are presented in Table 2.7.

2.7 Objective 3: To Develop a Method Capable of Adapting to the Dynamic Nature of Social Networks and the Complex Changes in Bot Features Over Time

OSNs have become a fundamental part of modern interactions and information exchange. With the increasing popularity of social networks, there has been an increase in the use of bots for various purposes, such as spamming, phishing, and disseminating false information. Therefore, bot detection techniques are employed to detect and eliminate bot accounts. However, the dynamic nature of social networks and ongoing growth in bot characteristics pose substantial obstacles to bot detection systems.

The dynamic nature of social networks means that they are continually evolving owing to the constant occurrence of new users, posts, and connections. This makes it challenging for bot detection algorithms with numerous features to adapt to the constantly shifting OSN landscape. As new users join a network, bots may be generated,

and the behavior or capabilities of existing bots may change. This implies that bot detection algorithms must be capable of adapting to these dynamic changes and accurately identifying bots despite these alterations.

To handle dynamic changes, we applied TL to our pretrained model (Bot-MGAT) from our previous study [106]. We applied the PAC learning framework to ensure that our model could make accurate in-stream predictions. The details are provided in Publication 3 in Publications.

2.7.1 Objective 3: Method

TL is an ML technology that uses the knowledge of previously trained models (features, weights, etc.) to train new models. It addresses the problem of insufficient training data for learning models [129]. There are three types of TL: inductive, unsupervised, and transductive. Inductive TL is applied when labels exist only in the target domain and not in the source domain, and both the source and destination tasks are related yet distinct. Unsupervised TL is employed when there are no labels in either the source or the target domains. Transductive TL is employed when source labels are accessible but destination labels are absent [130].

In our case, both the source and target tasks included Twitter datasets; thus, there were parallels. The source dataset was constructed using friendship relationships (i.e., account A follows account B), whereas the target dataset was constructed using interaction links. In this scenario, the source domain contained enough labeled data, whereas the target domain contained no labeled data. As shown in Figure 2.18, we used transductive TL in our study. We trained our model using TwiBot-20 (source), which comprises many types of bots, to transfer as much knowledge as possible to differentiate between bot and human classes in the incoming data (target). We used two datasets that were never used for training (climate hashtag and Russia hashtag), as shown in Table 2.1. Given the rise of sophisticated bots that can resist detection, employing this method made our bot detection model more generic.

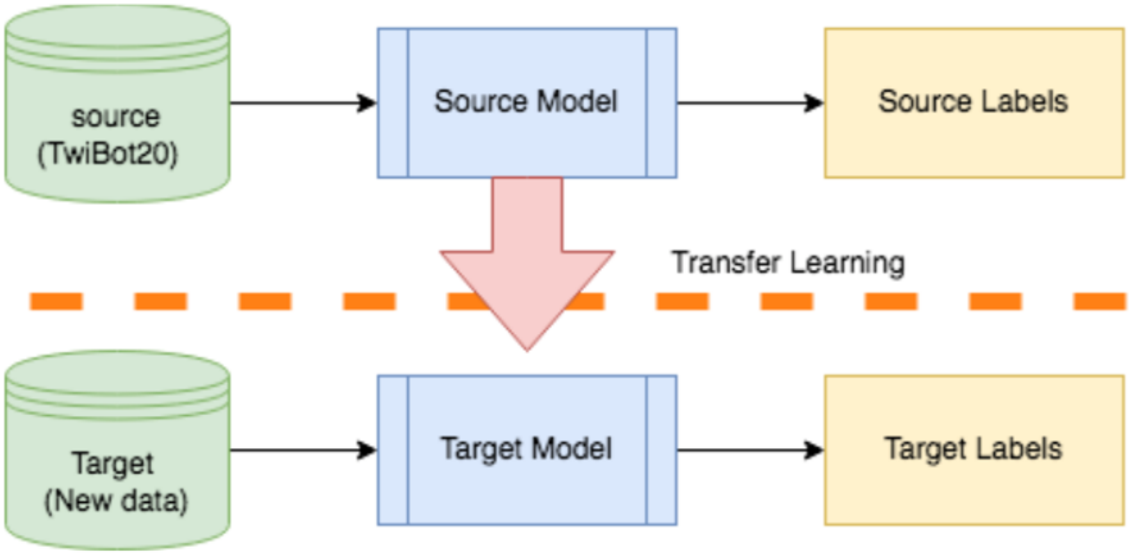


Figure 2.18: Transfer learning approach

Additionally, to ensure that our model could accommodate the dynamic nature of Twitter streams, we applied PAC learning, which is a mathematical analytical framework [131]. The SEBD framework was evaluated using PAC learning parameters to ensure the accuracy and confidence levels of the predictions. A PAC learner attempts to learn a concept (about correctness) by selecting a hypothesis with a low generalization error from a set of hypotheses. In the context of ML, a problem is PAC learnable if there exists an algorithm that, given some independently drawn samples, provides a hypothesis with a small error and high probability for any distribution and concept. However, developing a perfect hypothesis with zero error is impossible. Thus, the objective is to develop a consistent hypothesis that can make approximately accurate predictions with an upper error bound.

2.7.2 Objective 3: Result

The results of the TL implementation for bot detection using the Bot-MGAT model are presented in Table 2.8. It is obvious that the multi-view GAT with TL makes Bot-MGAT outperform other models as it allows uncovering hidden node representations to apply to the target using the knowledge learned from the source. Furthermore, these results indicate that TL is crucial for improving the model. While attempting to detect bot accounts in a particular domain, it is recommended to use a

training dataset that is both recent and rich in samples from a variety of bot domains (e.g., hashtags). Twibot-20 contains accounts from four domains: politics, business, entertainment and sports.

Table 2.8: Performances of Bot-MGAT and baselines

REF#	Dataset	Approach	Accuracy	F1 Score	MCC
BotRGCN [59]	TwiBot-20	RGCN	0.8462	0.8707	0.7021
Alhosseini et al. [71]	Yang C et al. [132] 2013	GCNN	0.94	0.84	-
Bot-MGAT$_{no_TL}$	TwiBot-20	MGAT	0.8098	0.8485	0.6491
Bot-MGAT$_{TL}$	TwiBot-20 + Climate hashtag		**0.9558**	**0.9739**	**0.8334**
Bot-MGAT$_{TL}$	TwiBot-20 + Russian hashtag		**0.9780**	**0.9842**	**0.9481**

For the PAC learning application, we tested our predicted samples by RF with the PAC framework parameters. The results are presented in Table 2.9, which indicate that the FPRs and FNRs for the generated sample sizes are less than the epsilon (0.05) and delta (0.05) thresholds. The range of minimum and maximum values for FNRs and FPRs with each sample size shows that the model can maintain values lower than the thresholds for data samples with more than 2000 data points. The more data it processes, the more accurate and confident its performance. In the empirical analysis of the model using PAC learning, we compared the true and empirical errors with the upper-bound generalization error (worst-case). Figure 2.19 shows the results. In addition, we verified the performance based on the model estimators (number of trees) versus accuracy, as shown in Figure 2.20. These findings indicate that the model performed better on unseen data than on the training data. This could be because the model learned more generalizable patterns from the training data and was overfitted to it, which contributed to its performance improvements for unseen data. In general, this indicates that the model performed well for bot account detection.

Table 2.9: Sample size ranges with minimum and maximum FNRs and FPRs for the group 2 dataset with PAC parameters

Sample size	FNRs	FPRs
50-2000	[0.024, 0.038]	[0.005, 0.196]
2050-5000	[0.033, 0.035]	[0.001, 0.031]
5050-10000	[0.034, 0.035]	[0.001, 0.023]
> 10050	[0.034, 0.035]	[0.001, 0.021]

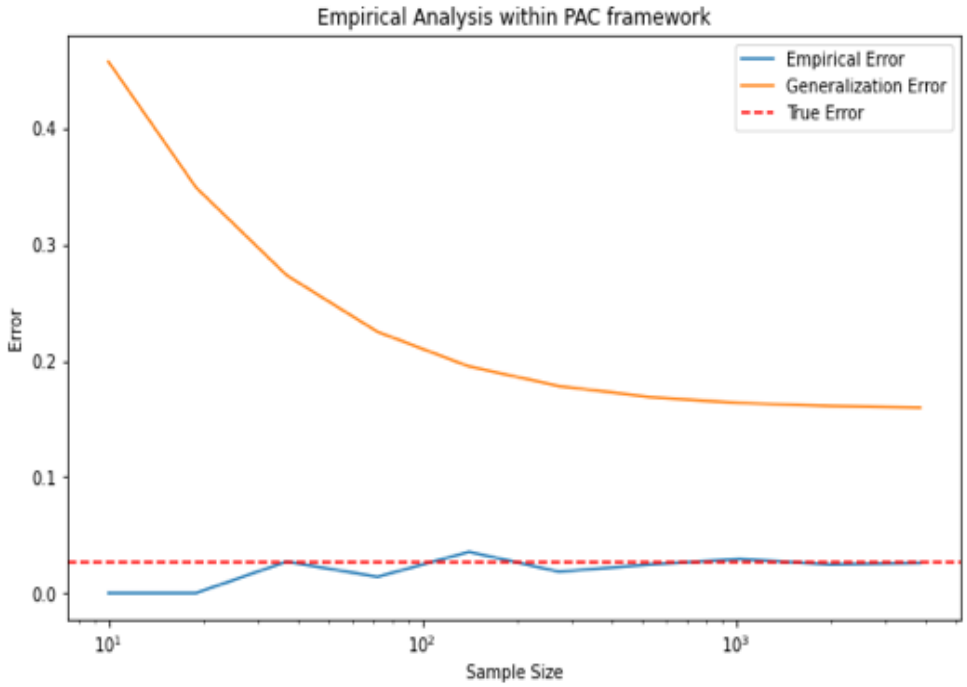


Figure 2.19: Empirical analysis using PAC framework for SEBD

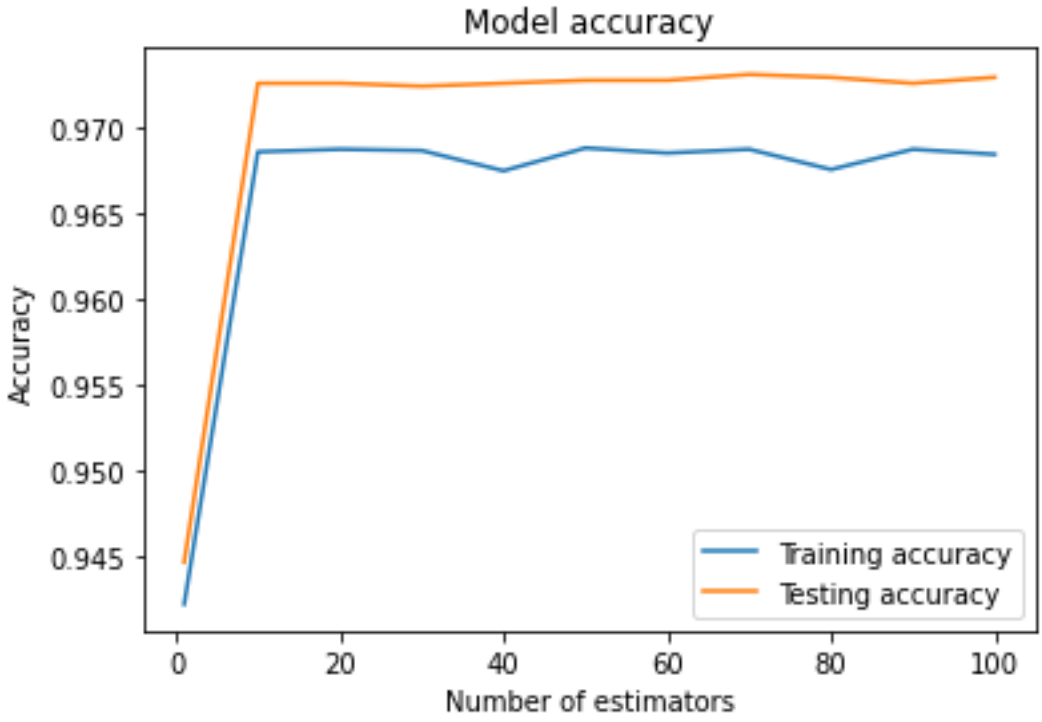


Figure 2.20: Model accuracy according to the number of estimators

Chapter 3: Findings & Discussion

The research conducted for this PhD thesis has yielded a number of significant findings. In this section, the key findings from the research are presented and discussed.

3.1 Dataset Size

The size of the labeled data is crucial for any ML algorithm. The good performance of the SEBD framework indicates that the trained dataset was effective. Obviously, the more labeled the samples, the more accurate the model. In our case, Twibot-20 was the only recently available dataset with follower/following links that we could use. Therefore, to investigate and understand social bot behavior and compare it with human activity on OSNs, it is critical to maintain recent datasets that include both human and bot accounts. Hence, researchers face two challenges when using these datasets. The first is the availability of recent public datasets for testing. Several studies used their own platforms to collect data to avoid this issue and Twitter's limit of one API request per hour. Unfortunately, this approach typically entails considerable time to acquire a dataset of sufficient size and enrich its content. In this regard, a significant number of studies include links for their processed datasets, which might be useful for comparison or testing learning models [61, 113]. However, to avoid violating user privacy, the characteristics and sizes of such datasets are limited, although they include IDs and labels to allow researchers to retrieve other features using the Twitter API. However, because of the difference in retrieval time and usage, retrieving features of many accounts that are classified as bots might not be possible as they have been suspended or their usernames have been changed to avoid suspension. This typically results in imbalanced data, as in [110].

Another challenge is obtaining a dataset with different contents of bot accounts that have been trained, given that many studies employ ML. Tagging a sample with well-defined size and composition is difficult. Thus, most datasets rely on human annotation with enough training samples to accomplish this task, even though this process is time-consuming and prone to human mistakes. To address this problem, some studies have considered suspended Twitter accounts as bot accounts. Nevertheless, this method is not particularly precise as human users are also occasionally suspended from violating

Twitter's terms of service. Additionally, it requires acquiring suspended account history that is not always accessible.

3.2 Profile Features for Detecting Social Bots

As mentioned in Section 2.5, feature selection can substantially affect the performance of bot detection models, because irrelevant or redundant features can result in overfitting, degrade model interpretability, and increase computing complexity. Profile features, including username, profile image, bio, description, location, and URL, have been used in most bot detection methods. These models also include statistical features, such as the number of friends, tweets, and followers. Experiments have shown that bots tend to adjust their features to avoid discovery. Thus, profile features can be predictive, as indicated in [133]. Table 3.1 lists the selected features for the SEBD and their descriptions.

Table 3.1: Descriptions of features selected for SEBD

Feature	Type	Description
Follower count	Numerical	The number of users who are following this user account
Favorites count	Numerical	The number of tweets a user has liked since the account creation date
Status count	Numerical	Total number of tweets and retweets made by the user
Friend count	Numerical	Total number of accounts the user is following
Age in days	Numerical	Total number of days since the account was created until the day of data retrieval
Screen name length	Numerical	Total number of digits in the screen name
Verified	Boolean	Indication that the user has a verified account
Default profile	Boolean	TRUE if the user has not altered the theme or background of their profile
Default profile image	Boolean	TRUE if the user has not uploaded a profile image and the default image is used
Geo-enabled	Boolean	TRUE if the user includes geographic data when tweeting or retweeting

In our case, we found that profile features were predictive, as shown by the model, for two main reasons. First, we used a graph-based model to identify bots by using the neighboring representation to classify a given node. Therefore, bot accounts that are a part of a botnet have similar profile features because botmasters usually generate bot accounts in groups and not individually. Therefore, many of them have

similar profile features, which aid their discovery using GAT models. Thus, profile features are predictive for identifying such similarities and classifying such accounts as bots.

Second, several profile features are considered human features, such as verification options, as only a few bot accounts are verified. These accounts are usually newsfeeds or business-automated answering machines. Therefore, recent changes in Twitter policy that have made the verification option accessible to any user by paying $8[3] might affect the importance of this feature in the future (Figure 3.1).

In addition, we compared the feature distribution among the classes; for example, screen name length and age of accounts for bots and humans in different datasets. We found that despite the different collection times and processing of these datasets, they were similar when comparing such features, as shown in Figure 3.2.

[3] https://help.twitter.com/en/using-twitter/twitter-blue (accessed March 10th 2023)

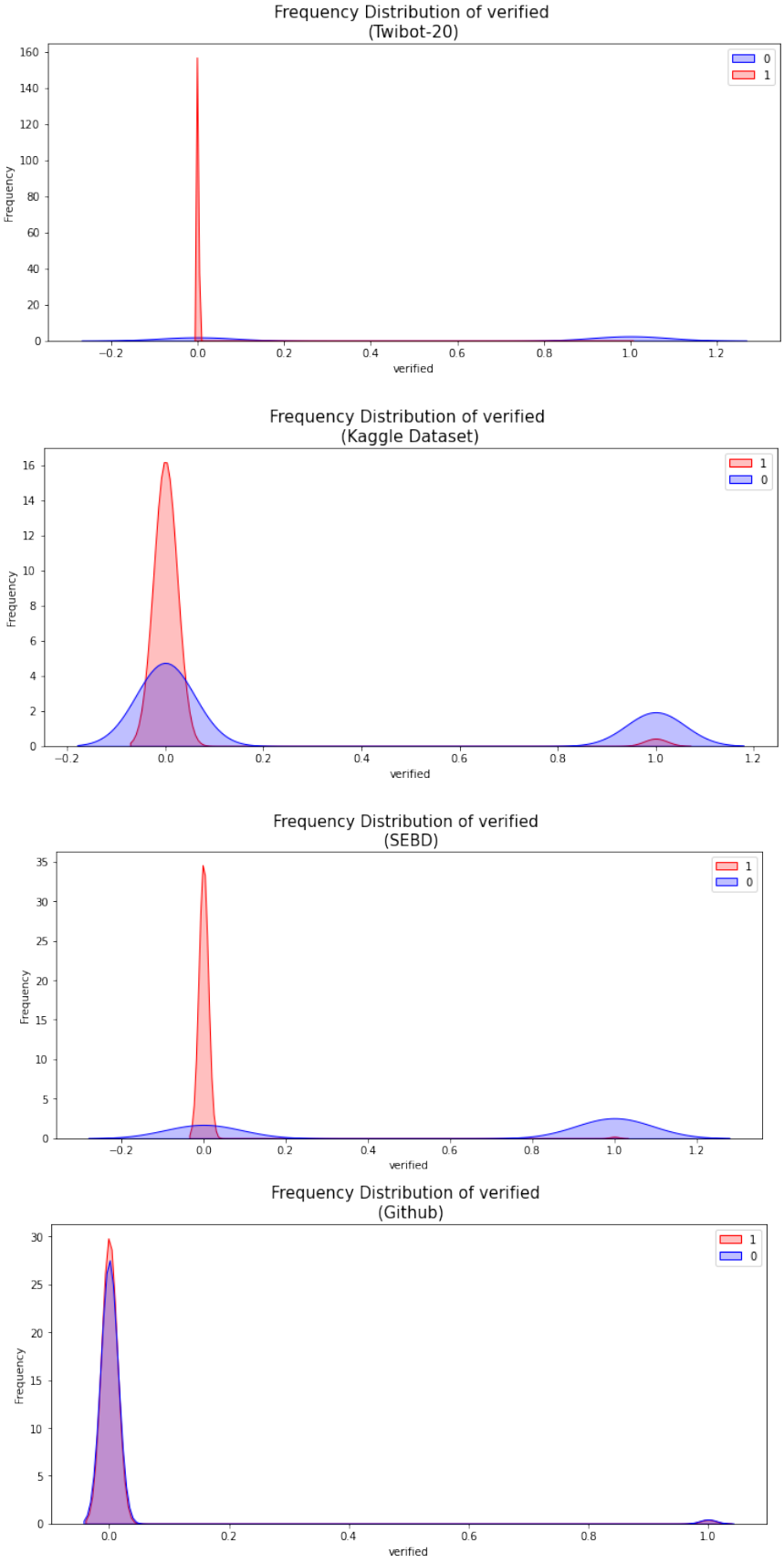


Figure 3.1: Frequency distributions of the verified feature across all datasets

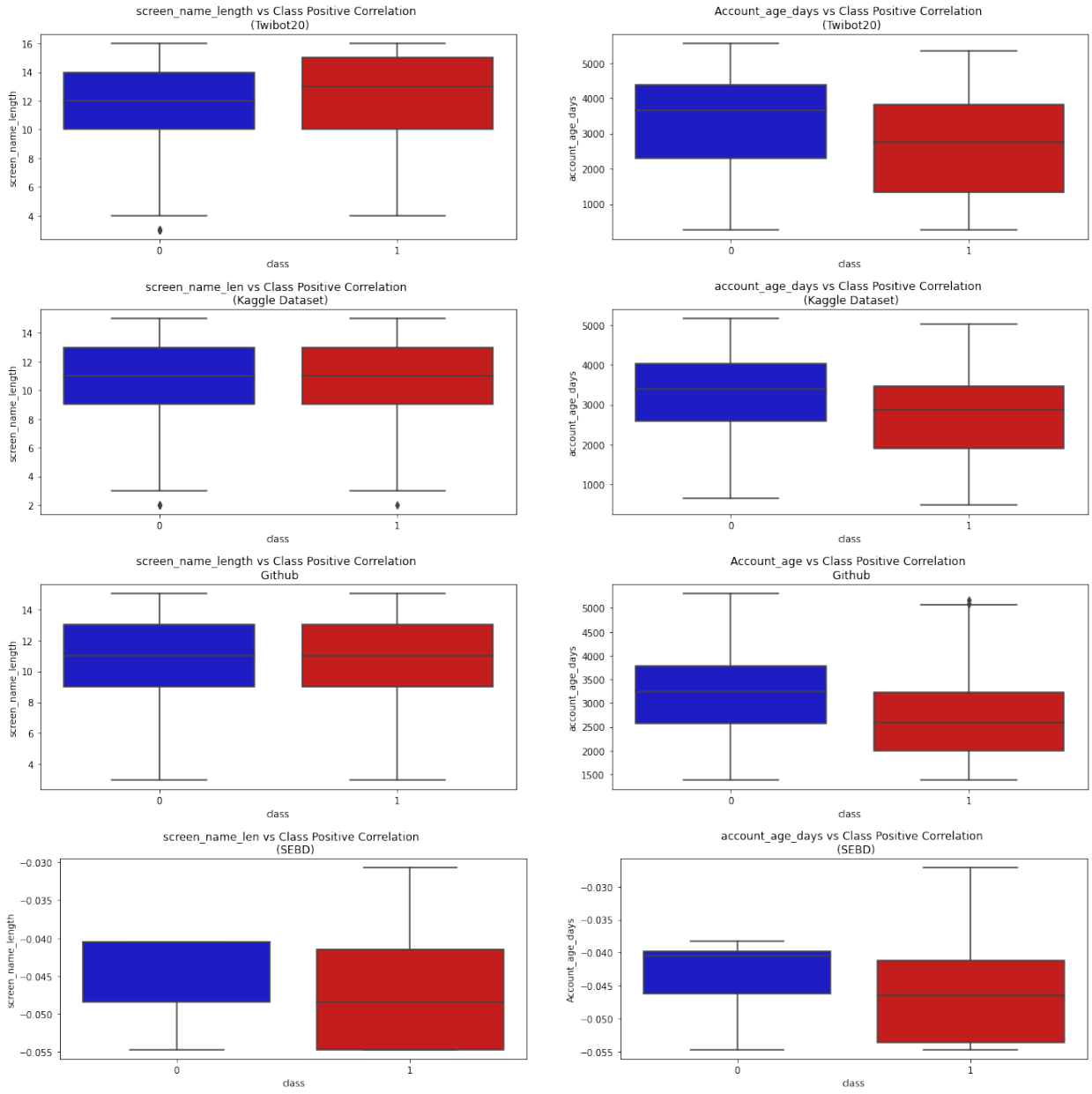


Figure 3.2: Screen name lengths and account ages in the Twibot20, Kaggle, Github, and SEBD datasets

3.3 Stream Bot Detection

The performance results of the SEBD framework indicate that it can perform on-stream detection of social bots. The results showed that it obtained higher recall scores than precision scores. This indicates that the SEBD can detect social bots. However, the precision scores indicate that a few human accounts were misclassified as bots. This is

because bot accounts tend to follow human accounts to avoid detection. Figure 3.3 illustrates the t-distributed stochastic neighbor embedding (T-SNE), an effective method for visualizing high-dimensional data used for dimensionality reduction without altering the structure of the data. The scatter plot highlights clusters of connected data points (red represents bots, and blue represents human accounts) and can be used to identify patterns and trends in the data. Clearly, bots (red colored) have fellowship links with human (blue colored) accounts to hide themselves. These results are consistent with those of many previous studies [91, 118, 134]. Such behavior affects the classification of human accounts if the number of bot followers is high. In addition, these accounts could be cyborg accounts (human accounts with automation features), as they have some features similar to those of bots [128].

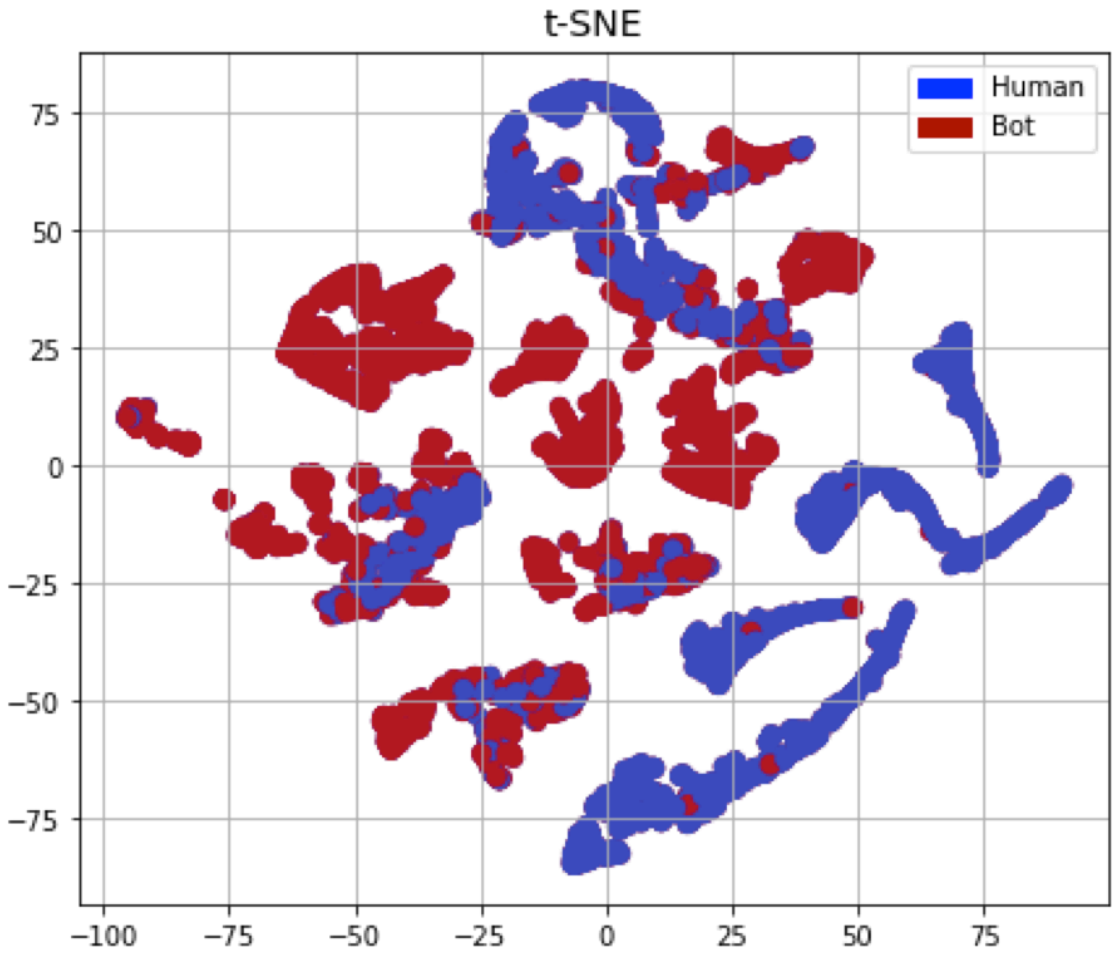


Figure 3.3: Visualization of SEBD results using t-SNE

Accordingly, it is important to note that the predictive features selected for SEBD play an important role. The preprocessing of these features and the normalization step accelerates the process to distinguish between accounts. Therefore, the pretrained Bot-MGAT model could identify such accounts by using these features to classify nodes based on neighboring embeddings. As shown in Figure 3.3, detecting social bots while they are evolving into hashtags is effective. This is because they are probably parts of botnets (clusters) and have similar features, making them distinguishable. Additionally, bot accounts are highly active in hashtags, as reported in recent studies [8, 37, 133].

Overfitting can be an issue when there is only a small variance between the training and cross-validation; however, this is not always the case [135]. It is possible that the small variance is owing to noise in the data, and the model still performs well. In this case, the model is not overfitted, and the small variance is not problematic. Therefore, when deciding whether overfitting is a problem, it is crucial to consider the context and data as indicated by the SEBD results. In addition, the RF and ensemble methods are nonparametric, which means that they make no assumptions about the underlying data distribution. This makes them more adaptable and capable of capturing complicated correlations, which are useful for detecting bots whose behaviors are unpredictable. Therefore, many studies have integrated RF classifiers into their bot detection systems [41, 50, 86, 136].

Imbalance is another issue in bot detection data. SEBD has numerous bot predictions. By contrast, the Kaggle dataset contains a few bots. Therefore, this issue must be addressed to ensure accurate predictions. To ensure the robustness of our model, we used an undersampling method to address imbalance issues in any dataset, including the SEBD. Thus, we wanted to ensure that the data were not skewed and the predictions were accurate. We applied cross-validation using several ML algorithms. We can clearly observe that the SEBD generally outperforms the other datasets. This emphasizes that multi-view GATs can be powerful tool for detecting bot accounts in streams. Figure 3.4 - 3.7 show the undersampling results for SEBD, Twibot-20, kaggle.com, and Github datasets, respectively.

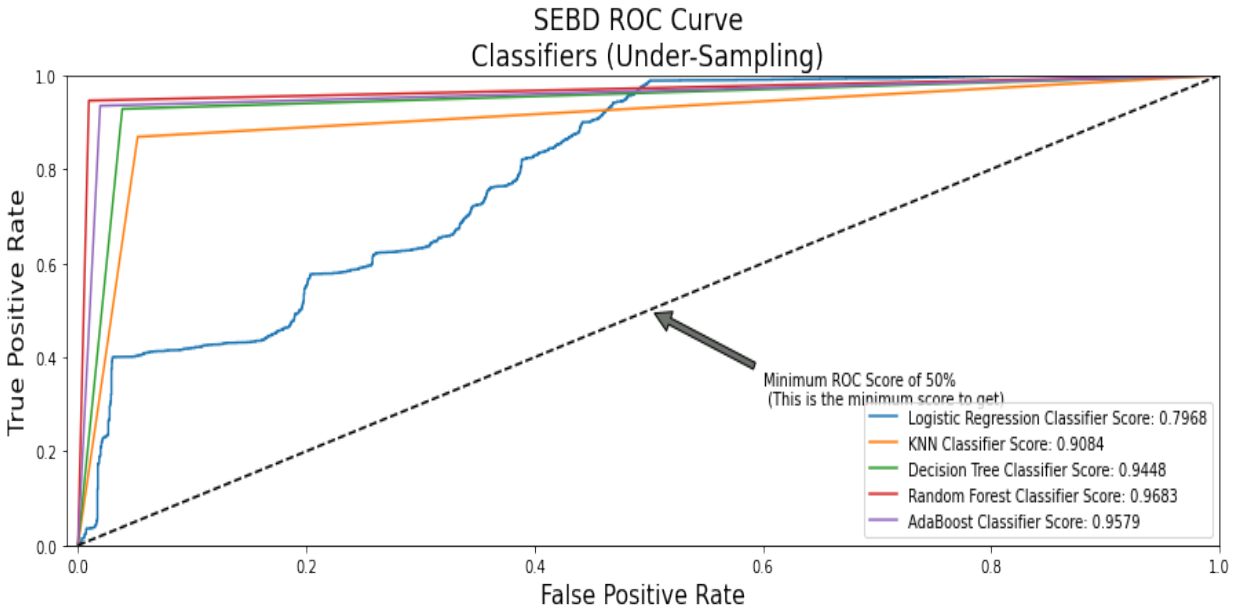


Figure 3.4: AUCs of undersampling algorithms for the SEBD dataset

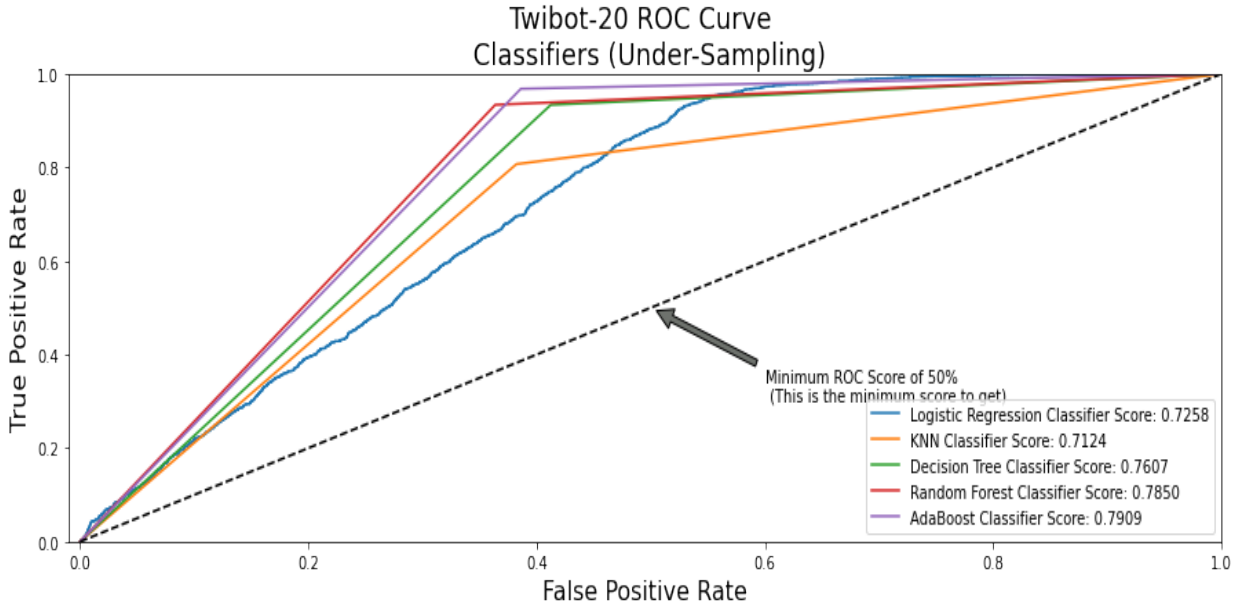


Figure 3.5: AUCs of undersampling algorithms for the Twibot-20 dataset

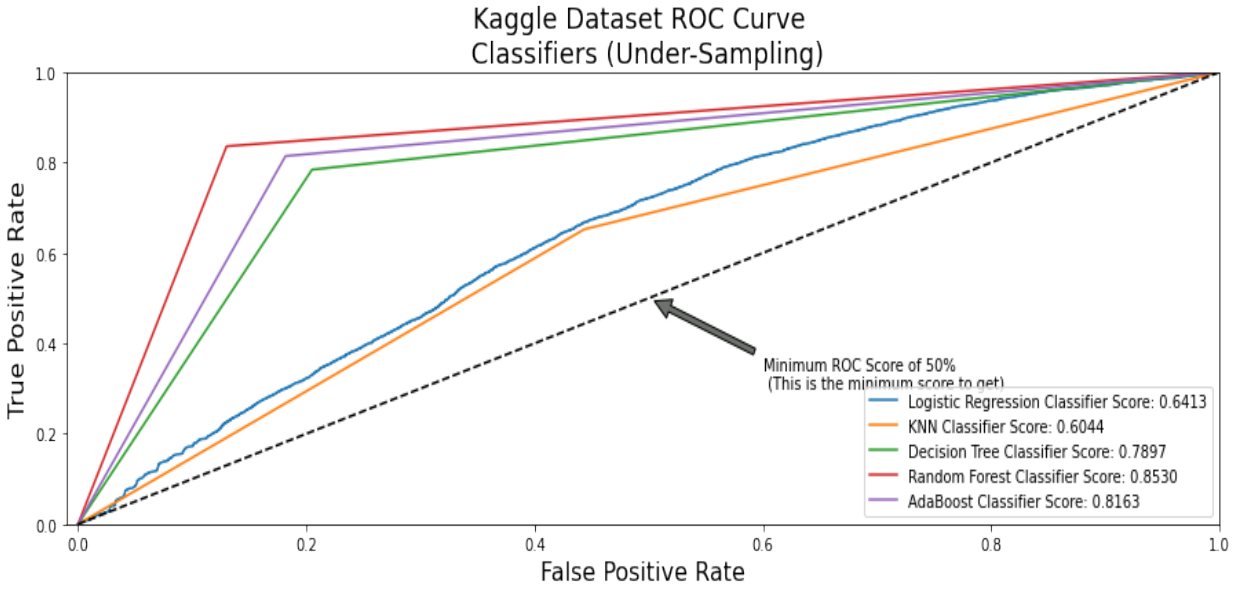


Figure 3.6: AUCs of undersampling algorithms for the Kaggle dataset

t

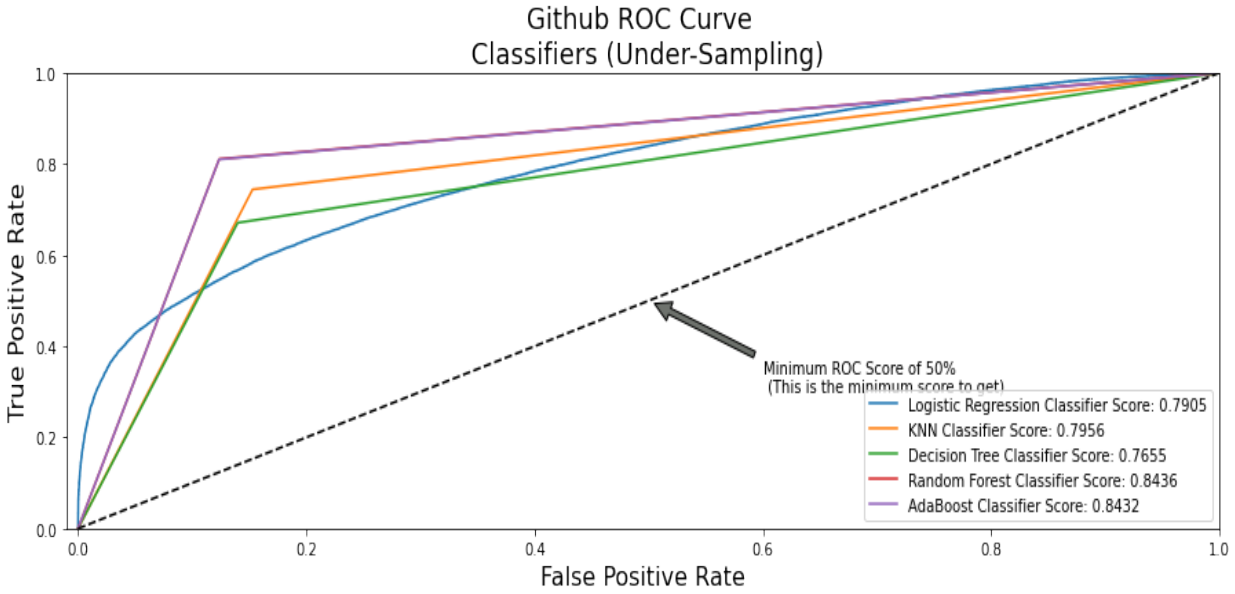


Figure 3.7: AUCs of undersampling algorithms for the Github dataset

3.4 Handling the Dynamic Nature of Social Networks

Artificial intelligence (AI) has played an important role in the revolution of digital assistance devices and chatbot applications. This revolution has resulted in the creation of bots that mimic human behavior, which has increased their ability to avoid detection. The performance results of the SEBD framework show that it can successfully identify

bot accounts. It can handle the dynamics of Twitter streams, as demonstrated through different evaluation performance measures and comparisons with baselines. This finding is supported by the application of the PAC parameter framework to ensure the accuracy and confidence of the model predictions. The results show that the SEBD should have at least 2000 samples to maintain accuracy and confidence levels in terms of FPR and FNR. This is because the SEBD framework was built using 10 predefined profile features. The values of these features are numerical, making the processing straightforward. The challenge is that if SEBD handles tweets, the dimensions of these features increase, and a handling approach is required.

Accordingly, we found that TL is effective for handling Twitter dynamics, as shown in our previous study [106]. TL can be used to manage dynamic changes in data. We tested this approach using different graph links and views (fellowship and interaction) and the results were found to be superior compared to single view or no view in GAT model. In reality, TL is highly useful if the target task has a limited amount of data or the data distribution of the target task differs significantly from that of the source task. TL allows the model to adapt more rapidly and accurately to new data by using the knowledge it learns from an existing dataset. This can be helpful in some situations, such as when data are scarce, as in bot detection problems, or are changing quickly.

In conclusion, TL is a powerful method for adjusting previously trained models to different input types [137]. It is used for enhancing the effectiveness and efficiency of ML models for different tasks and data distributions by leveraging the knowledge obtained from the source task.

3.5 Research Limitations

This study had some limitations that must be highlighted. First, we trained our model (Bot-MGAT) using only one dataset (Twibot20). Despite the accuracy of our model, this may be an issue. However, as previously mentioned, the shortage of labeled data with graph links is a critical problem. Using a single dataset may not be typical for the entire bot population, particularly if it is collected during a particular period or by using a particular methodology. This could introduce bias and reduce the generalizability

of the model. To overcome this limitation, in a future enhancement of our model, we will include a newly released dataset [138].

Additionally, there is a tradeoff between reducing computational costs and the number of selected features in the SEBD framework. Despite its competitive performance, relying exclusively on profile characteristics may not allow it capture the interaction behavior of bots when they engage with other users. Bots may alter their behavior, making it more difficult to identify them based solely on profile characteristics. Therefore, we will consider adding interaction features to address this problem.

Another limitation that needs to be addressed in the future are the constraints set by the Twitter API on the volume and frequency of data that may be retrieved, which can hinder in-stream bot detection. For example, the standard API restricts the number of requests in each 15-minute window, thereby restricting the amount of data that can be collected in real time. In addition, the API may introduce bias into the data collection procedure because its search and filtering algorithms may prefer particular types of information or users. This is because Twitter allows only 1% of its data to be collected [139]. To circumvent these limitations, researchers and developers may utilize alternative data sources and methods, such as web scraping, crowdsourcing, and Twitter collaboration, to access extra data. Several of these methods may be subject to legal or ethical considerations and should be used with caution.

Binary classification of bot detection can classify human accounts with similar behavior like bots as bots account. This might lead to account suspension if not addressed well. Therefore, considering three classes might be more effective as we have some human accounts that are cyborg.

Finally, we designed and tested our model using Twitter as the OSN. Therefore, the framework must be adapted to detect bots on other OSNs owing the differences in their graph structures and links. In addition, user features vary across OSNs, which can play a critical role in the model performance.

Chapter 4: Conclusions and Future Perspectives

This dissertation addressed the problem of detecting bots on OSNs, particularly Twitter. Detecting bot accounts has become a challenging task, particularly with the advancement of AI. This research focused on detecting bots in stream while they evolve into a hashtag, as this process is more effective than detecting them randomly. The SEBD framework was built using a multi-view GAT with fellowship links and a few profile features to identify bot accounts. Its performance was evaluated using multiple approaches to ensure the accuracy and confidence of its predictions. The results showed that the SEBD framework can effectively identify bot accounts and maintain accuracy and confidence levels with the PAC framework parameters.

Our findings indicate that profile features are optimal for identifying social bots, as indicated in other studies [51, 59, 61]. The multi-view GAT model outperformed other models for bot detection. It uses fellowship links and reveals complex relationships between network nodes to predict the labels of neighboring nodes using node embeddings. Our results indicate that in-stream detection of bot accounts shows promising results compared to offline detection. However, recent diverse large datasets must be updated with links to maintain the model performance. The results show that Twitter may have more bots as users than humans or that these accounts might be cyborg accounts. Our results indicate that bots participate significantly in hashtags, which is supported by a recent study [69].

4.1 Research Implications

In general, many aspects and far-reaching implications of bots make bot detection research on Twitter crucial. The key points are as follows:

1. Maintaining the integrity of social media. Twitter and other social media platforms have evolved into potent tools for social organizations, information exchange, and communication. Researchers can contribute to preserving the credibility of social media as a legitimate information source by creating efficient methods of identifying and eliminating bots.

2. Bot accounts can be used to perform cyberattacks, spread malware, and steal confidential information, as mentioned in Section 1.4. By detecting and eliminating bots, researchers can help prevent security breaches and safeguard users.
3. Academics commonly mine social media for data to investigate patterns, assess public sentiment, or analyze the distribution of ideas. If bots generate a considerable portion of these data, the conclusions may be invalid and the reliability of the research may be jeopardized.
4. Efficient bot detection systems can assist in mitigating negative effects by identifying and removing accounts intending to spread misinformation and rumors on OSNs.
5. Investigating bot behavior on Twitter and other platforms can provide insights into the techniques used by malicious actors to manipulate public opinion and spread misinformation.

4.2 Future Work

Bot detection is challenging because bots adapt their features. Therefore, there is room for improvement and enhancement. This is because as they have evolved, bot accounts have become adaptive by using AI to avoid detection mechanisms. Thus, in the future, the SEBD framework will be improved in the following areas:

- Interaction links will be added in the stream as an additional view for the pretrained model (Bot-MGAT).
- Tweet content analysis to be added to identify the behavior and context of bots.
- The limit on data retrieval must be increased (a request to Twitter) and a solution for dropping out of connection by utilizing our own API to launch our tool for academic public use must be found.
- Enhancing the capacity to process more data using advanced hardware to enlarge the graph.

References

[1] Latah M. Detection of malicious social bots: A survey and a refined taxonomy. *Expert Syst Appl* 2020; 151: 113383.

[2] Ferrara E, Varol O, Davis C, et al. The rise of social bots. *Commun ACM* 2016; 59: 96–104.

[3] Alothali E, Zaki N, Mohamed EA, et al. Detecting Social Bots on Twitter: A Literature Review. In: *Proceedings of the 2018 13th International Conference on Innovations in Information Technology, IIT 2018*. IEEE, pp. 175–180.

[4] Cresci S. A decade of social bot detection. *Commun ACM* 2020; 63: 72–83.

[5] Varol O, Ferrara E, Davis CA, et al. Online human-bot interactions: Detection, estimation, and characterization. *Proc 11th Int Conf Web Soc Media, ICWSM 2017* 2017; abs/1703.0: 280–289.

[6] Smith C. 400 Amazing Twitter Statistics and Facts, https://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats (2017). Accessed on: 30th November 2017

[7] Abokhodair N, Yoo D, McDonald DW. Dissecting a social botnet: Growth, content and influence in Twitter. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. 2015, pp. 839–851.

[8] Chen C-F, Shi W, Yang J, et al. Social bots' role in climate change discussion on Twitter: Measuring standpoints, topics, and interaction strategies. *Adv Clim Chang Res* 2021; 12: 913–923.

[9] Subrahmanian VS, Azaria A, Durst S, et al. The DARPA Twitter Bot Challenge. *Computer (Long Beach Calif)* 2016; 49: 38–46.

[10] Grier C, Thomas K, Paxson V, et al. @ spam: the underground on 140 characters or less. In: *Proceedings of the 17th ACM conference on Computer and communications security*. 2010, pp. 27–37.

[11] Stringhini G, Kruegel C, Vigna G. Detecting spammers on social networks. In: *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*. New York, New York, USA: ACM Press, p. 1.

[12] Wang AH. Detecting spam bots in online social networking sites: A machine learning approach. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, pp. 335–342.

[13] Zhang X, Zhu S, Liang W. Detecting spam and promoting campaigns in the Twitter social network. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*. IEEE, pp. 1194–1199.

[14] Rathore S, Sharma PK, Loia V, et al. Social network security: Issues, challenges, threats, and solutions. *Inf Sci (Ny)* 2017; 421: 43–69.

[15] Shafahi M, Kempers L, Afsarmanesh H. Phishing through social bots on Twitter. In: *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*. IEEE, pp. 3703–3712.

[16] Zhang J, Zhang R, Zhang Y, et al. The rise of social botnets: Attacks and countermeasures. *IEEE Trans Dependable Secur Comput*.

[17] Rizoiu MA, Graham T, Zhang R, et al. DEBATENIGHT: The role and influence of socialbots on twitter during the first 2016 U.S. presidential debate. In: *12th International AAAI Conference on Web and Social Media, ICWSM 2018*. 2018, pp. 300–309.

[18] Keller TR, Klinger U. Social Bots in Election Campaigns: Theoretical, Empirical, and Methodological Implications. *Polit Commun* 2019; 36: 171–189.

[19] Ferrara E, Cresci S, Luceri L. Misinformation, manipulation, and abuse on social media in the era of COVID-19. *J Comput Soc Sci* 2020; 3: 271–277.

[20] Yang KC, Varol O, Davis CA, et al. Arming the public with artificial intelligence to counter social bots. *Hum Behav Emerg Technol* 2019; 1: 48–61.

[21] Lou X, Flammini A, Menczer F. Information Pollution by Social Bots. *arXiv Prepr arXiv190706130*.

[22] Gupta A, Lamba H, Kumaraguru P. $1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on twitter. In: *eCrime Researchers Summit (eCRS), 2013*. 2013, pp. 1–12.

[23] Twitter Inc. About Twitter Blue. *Twitter.com*, https://help.twitter.com/en/using-twitter/twitter-blue (2022, accessed 30th December 2022).

[24] Lee BY, Follow SC. Fake Eli Lilly Twitter Account Claims Insulin Is Free, Stock Falls 4.37% Listen to article 8 minutes. *Forbes.com*, https://www.forbes.com/sites/brucelee/2022/11/12/fake-eli-lilly-twitter-account-claims-insulin-is-free-stock-falls-43/ (2022, accessed 30 December 2022).

[25] Cresci S, Lillo F, Regoli D, et al. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. *ACM Trans Web* 2019; 13: 11.

[26] Boshmaf Y, Muslukhov I, Beznosov K, et al. Design and analysis of a social botnet. *Comput Networks* 2013; 57: 556–578.

[27] Cresci S, Di Pietro R, Petrocchi M, et al. Fame for sale: efficient detection of fake Twitter followers. *Decis Support Syst* 2015; 80: 56–71.

[28] Amato F, Castiglione A, De Santo A, et al. Recognizing human behaviours in online social networks. *Comput Secur*.

[29] Sivanesh S, Kavin K, Hassan AA. Frustrate Twitter from automation: How far a user can be trusted? In: *Human Computer Interactions (ICHCI), 2013 International Conference on*. 2013, pp. 1–5.

[30] Laboreiro G, Sarmento L, Oliveira E. Identifying automatic posting systems in microblogs. *Prog Artif Intell* 2011; 634–648.

[31] Zhang CM, Paxson V. Detecting and Analyzing Automated Activity on Twitter. In: Spring N, Riley GF (eds) *Passive and Active Measurement: 12th International Conference, PAM 2011, Atlanta, GA, USA, March 20-22, 2011. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 102–111.

[32] Chavoshi N, Hamooni H, Mueen A. Temporal Patterns in Bot Activities. In: *Proceedings of the 26th International Conference on World Wide Web Companion*. 2017, pp. 1601–1606.

[33] Fu Q, Feng B, Guo D, et al. Combating the evolving spammers in online social networks. *Comput Secur* 2018; 72: 60–73.

[34] Lee S, Kim J. Early filtering of ephemeral malicious accounts on Twitter. *Comput Commun* 2014; 54: 48–57.

[35] NaliniPriya G, Asswini M. A survey on vulnerable attacks in online social networks. In: *International Confernce on Innovation Information in Computing Technologies*. 2015, pp. 1–6.

[36] Lee K, Eoff BD, Caverlee J. Seven months with the devils: A long-term study of content polluters on twitter. In: *Fifth International AAAI Conference on Weblogs and Social Media*. 2011.

[37] Yuan X, Schuchard RJ, Crooks AT. Examining Emergent Communities and Social Bots Within the Polarized Online Vaccination Debate in Twitter. *Soc Media + Soc* 2019; 5: 2056305119865465.

[38] Broniatowski DA, Jamison AM, Qi S, et al. Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *Am J Public Health* 2018; 108: 1378–1384.

[39] Boshmaf Y, Muslukhov I, Beznosov K, et al. The socialbot network: when bots socialize for fame and money. In: *Proceedings of the 27th annual computer security applications conference*. 2011, pp. 93–102.

[40] Freitas C, Benevenuto F, Ghosh S, et al. Reverse engineering socialbot infiltration strategies in twitter. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 2015, pp. 25–32.

[41] Yang K-C, Varol O, Hui P-M, et al. Scalable and Generalizable Social Bot Detection through Data Selection. *Proc AAAI Conf Artif Intell* 2020; 34: 1096–1103.

[42] Adewole KS, Anuar NB, Kamsin A, et al. Malicious accounts: Dark of the social networks. *J Netw Comput Appl* 2017; 79: 41–67.

[43] Davis CA, Varol O, Ferrara E, et al. Botornot: A system to evaluate social bots. In: *Proceedings of the 25th International Conference Companion on World Wide Web*. 2016, pp. 273–274.

[44] Alarifi A, Alsaleh M, Al-Salman A. Twitter turing test: Identifying social machines. *Inf Sci (Ny)* 2016; 372: 332–346.

[45] Kantepe M, Ganiz MC. Preprocessing framework for Twitter bot detection. In: *Computer Science and Engineering (UBMK), 2017 International Conference on*. 2017, pp. 630–634.

[46] Khalil A, Hajjdiab H, Al-Qirim N. Detecting Fake Followers in Twitter: A Machine Learning Approach. *Int J Mach Learn Comput* 2017; 7: 198–202.

[47] Najari S, Salehi M, Farahbakhsh R. GANBOT: a GAN-based framework for social bot detection. *Soc Netw Anal Min* 2022; 12: 1–11.

[48] Li G, Qiao Y, Zheng Y, et al. Semi-Supervised Learning Based on Generative Adversarial Network and Its Applied to Lithology Recognition. *IEEE Access* 2019; 7: 67428–67437.

[49] Wu B, Liu L, Yang Y, et al. Using Improved Conditional Generative Adversarial Networks to Detect Social Bots on Twitter. *IEEE Access* 2020; 8: 36664–36680.

[50] Rodrigues AP, Fernandes R, A A, et al. Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques. *Comput Intell Neurosci* 2022; 2022: 5211949.

[51] Kudugunta S, Ferrara E. Deep neural networks for bot detection. *Inf Sci (Ny)* 2018; 467: 312–322.

[52] Cresci S, Spognardi A, Petrocchi M, et al. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In: *26th International World Wide Web Conference 2017, WWW 2017 Companion*. 2017, pp. 963–972.

[53] Shevtsov A, Tzagkarakis C, Antonakaki D, et al. Explainable machine learning pipeline for Twitter bot detection during the 2020 US Presidential Elections. *Softw Impacts* 2022; 13: 100333.

[54] Feng S, Tan Z, Li R, et al. Heterogeneity-Aware Twitter Bot Detection with Relational Graph Transformers. *Proc AAAI Conf Artif Intell* 2022; 36: 3977–3985.

[55] Feng S, Wan H, Wang N, et al. TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark. *Int Conf Inf Knowl Manag Proc* 2021; abs/2106.1: 4485–4494.

[56] Mazza M, Avvenuti M, Cresci S, et al. Investigating the difference between trolls, social bots, and humans on Twitter. *Comput Commun* 2022; 196: 23–36.

[57] Hayawi K, Mathew S, Venugopal N, et al. DeeProBot: a hybrid deep neural network model for social bot detection based on user profile data. *Soc Netw Anal Min* 2022; 12: 43.

[58] Yang Y, Yang R, Li Y, et al. RoSGAS: Adaptive Social Bot Detection with Reinforced Self-Supervised GNN Architecture Search. *ACM Trans Web*. Epub ahead of print December 2022. DOI: 10.1145/3572403.

[59] Feng S, Wan H, Wang N, et al. BotRGCN: Twitter Bot Detection with Relational Graph Convolutional Networks. *arXiv Prepr arXiv210613092*.

[60] Cui W, Du J, Wang D, et al. MVGAN: Multi-view graph attention network for social event detection. *ACM Trans Intell Syst Technol*; 12. Epub ahead of print July 2021. DOI: 10.1145/3447270.

[61] Feng S, Wan H, Wang N, et al. {SATAR:} {A} Self-supervised Approach to Twitter Account Representation Learning and its Application in Bot Detection. *CoRR*; abs/2106.1, https://arxiv.org/abs/2106.13089 (2021).

[62] Shukla H, Jagtap N, Patil B. Enhanced Twitter bot detection using ensemble machine learning. In: *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*. IEEE, pp. 930–936.

[63] Martin-Gutierrez D, Hernandez-Penaloza G, Hernandez AB, et al. A Deep Learning Approach for Robust Detection of Bots in Twitter Using Transformers. *IEEE Access* 2021; 9: 54591–54601.

[64] Huang Q, Yu J, Wu J, et al. Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8.

[65] Sayyadiharikandeh M, Varol O, Yang KC, et al. Detection of Novel Social Bots by Ensembles of Specialized Classifiers. In: *International Conference on Information and Knowledge Management, Proceedings*. 2020, pp. 2725–2732.

[66] Mendoza M, Tesconi M, Cresci S. Bots in Social and Interaction Networks: Detection and Impact Estimation. *ACM Trans Inf Syst*; 39. Epub ahead of print October 2020. DOI: 10.1145/3419369.

[67] Mazza M, Cresci S, Avvenuti M, et al. RTbust: Exploiting Temporal Patterns for Botnet Detection on Twitter. *arXiv Prepr arXiv190204506*.

[68] Rodríguez-Ruiz J, Mata-Sánchez JI, Monroy R, et al. A one-class classification approach for bot detection on Twitter. *Comput Secur* 2020; 91: 101715.

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

[70] Cresci S, Pietro R Di, Petrocchi M, et al. Social Fingerprinting: Detection of Spambot Groups Through DNA-Inspired Behavioral Modeling. *IEEE Trans Dependable Secur Comput* 2018; 15: 561–576.

[71] Ali Alhosseini S, Bin Tareaf R, Najafi P, et al. Detect me if you can: Spam bot detection using inductive representation learning. In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 148–153.

[72] Amleshwaram AA, Reddy N, Yadav S, et al. Cats: Characterizing automation of twitter spammers. In: *Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on*. 2013, pp. 1–10.

[73] Badawy A, Lerman K, Ferrara E. Who Falls for Online Political Manipulation? In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 162–168.

[74] Stukal D, Sanovich S, Tucker JA, et al. For Whom the Bot Tolls: A Neural Networks Approach to Measuring Political Orientation of Twitter Bots in Russia. *SAGE Open* 2019; 9: 2158244019827715.

[75] Loyola-González O, Monroy R, Rodr$\$'$\$iguez J, et al. Contrast Pattern-Based Classification for Bot Detection on Twitter. *IEEE Access* 2019; 7: 45800–45817.

[76] Hafzullah I, Taner T. Interaction-Based Behavioral Analysis of Twitter Social Network Accounts. *Appl Sci* 2019; 9: 4448.

[77] Schuchard R, Crooks AT, Stefanidis A, et al. Bot stamina: examining the influence and staying power of bots in online social networks. *Appl Netw Sci* 2019; 4: 55.

[78] H. Nazer T, Davis M, Karami M, et al. Bot detection: Will focusing on recall cause overall performance deterioration? In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2019, pp. 39–49.

[79] Morstatter F, Wu L, Nazer TH, et al. A new approach to bot detection: Striking the balance between precision and recall. In: *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. 2016, pp. 533–540.

[80] Gilani Z, Farahbakhsh R, Tyson G, et al. A Large-scale Behavioural Analysis of Bots and Humans on Twitter. *ACM Trans Web* 2019; 13: 1–23.

[81] Balestrucci A, De Nicola R, Inverso O, et al. Identification of credulous users on Twitter. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 2019, pp. 2096–2103.

[82] Albadi N, Kurdi M, Mishra S. Hateful People or Hateful Bots? Detection and Characterization of Bots Spreading Religious Hatred in Arabic Social Media. *arXiv Prepr arXiv190800153*.

[83] Cresci S, Petrocchi M, Spognardi A, et al. Better Safe Than Sorry: An Adversarial Approach to Improve Social Bot Detection. *arXiv Prepr arXiv190405132*.

[84] Beskow DM, Carley KM. Its all in a name: detecting and labeling bots by their name. *Comput Math Organ Theory* 2019; 25: 24–35.

[85] De Cristofaro E, Kourtellis N, Leontiadis I, et al. LOBO: Evaluation of Generalization Deficiencies in Twitter Bot Classifiers. In: *Proceedings of the 34th Annual Computer Security Applications Conference*. 2018, pp. 137–146.

[86] Gilani Z, Farahbakhsh R, Tyson G, et al. Of bots and humans (on twitter). In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2017*. New York, NY, USA: ACM, pp. 349–354.

[87] Efthimion PG, Payne S, Proferes N. Supervised machine learning bot detection techniques to identify social twitter bots. *SMU Data Sci Rev* 2018; 1: 5.

[88] Jr SB, Campos GFC, Tavares GM, et al. Detection of Human, Legitimate Bot, and Malicious Bot in Online Social Networks Based on Wavelets. *ACM Trans Multimed Comput Commun Appl* 2018 Mar 26;14(1s):1-7.

[89] Cornelissen LA, Barnett RJ, Schoonwinkel P, et al. A network topology approach to bot classification. In: *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*. 2018, pp. 79–88.

[90] Stella M, Ferrara E, De Domenico M. Bots sustain and inflate striking opposition in online social systems. *arXiv Prepr arXiv180207292*.

[91] Gilani Z, Kochmar E, Crowcroft J. Classification of twitter accounts into automated agents and human users. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 2017, pp. 489–496.

[92] Cai C, Li L, Zengi D. Behavior enhanced deep bot detection in social media. In: *Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on*. 2017, pp. 128–130.

[93] Minnich A, Chavoshi N, Koutra D, et al. BotWalk: Efficient adaptive exploration of Twitter bot networks. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 2017, pp. 467–474.

[94] Jia J, Wang B, Gong NZ. Random Walk Based Fake Account Detection in Online Social Networks. In: *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*. 2017, pp. 273–284.

[95] Kwak H, Lee C, Park H, et al. What is Twitter, a social network or a news media? In: *Proceedings of the 19th International Conference on World Wide Web, WWW '10*. 2010, pp. 591–600.

[96] Chavoshi N, Hamooni H, Mueen A. DeBot: Twitter Bot Detection via Warped Correlation. In: *ICDM*. 2016, pp. 817–822.

[97] Mehrotra A, Sarreddy M, Singh S. Detection of fake Twitter followers using graph centrality measures. In: *Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on*. 2016, pp. 499–504.

[98] Zhang J, Zhang R, Sun J, et al. Truetop: A sybil-resilient system for user influence measurement on twitter. *IEEE/ACM Trans Netw* 2016; 24: 2834–2846.

[99] Lee K, Eoff BD, Caverlee J. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. In: *Icwsm 2011*. 2006, pp. 185–192.

[100] Chu Z, Gianvecchio S, Wang H, et al. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Trans Dependable Secur Comput* 2012; 9: 811–824.

[101] Cresci S, Di Pietro R, Petrocchi M, et al. DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection. *IEEE Intell Syst* 2016; 31: 58–64.

[102] Hamilton WL, Ying R, Leskovec J. Representation Learning on Graphs: Methods and Applications. *CoRR*; abs/1709.0, http://arxiv.org/abs/1709.05584 (2017).

[103] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks. *IEEE Trans neural networks Learn Syst* 2020; 32: 4–24.

[104] Xia F, Sun K, Yu S, et al. Graph Learning: A Survey. *arXiv Prepr arXiv210500696*.

[105] Alothali E, Hayawi K, Alashwal H. Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter. *Soc Netw Anal Min* 2021; 11: 1–15.

[106] Alothali E, Salih M, Hayawi K, et al. Bot-MGAT: A Transfer Learning Model Based on a Multi-View Graph Attention Network to Detect Social Bots. *Appl Sci*; 12. Epub ahead of print 2022. DOI: 10.3390/app12168117.

[107] Hui P-M, Yang K-C, Torres-Lugo C, et al. BotSlayer: real-time detection of bot amplification on Twitter. *J Open Source Softw* 2019; 4: 1706.

[108] Antenore M, Camacho-Rodriguez JM, Panizzi E. A comparative study of Bot Detection techniques methods with an application related to Covid-19 discourse on Twitter. *CoRR*; abs/2102.0, http://arxiv.org/abs/2102.01148 (2021).

[109] Orabi M, Mouheb D, Al Aghbari Z, et al. Detection of Bots in Social Media: A Systematic Review. *Inf Process Manag* 2020; 57: 102250.

[110] Martín Gutiérrez D. Twitter Bots Accounts. *Kaggle.com*. Epub ahead of print 2020. DOI: 10.34740/KAGGLE/DSV/1623389.

[111] Botometer. Datasets, https://botometer.osome.iu.edu/bot-repository/datasets.html (2020, accessed 2 July 2021).

[112] Yang K-C, Ferrara E, Menczer F. Botometer 101: Social bot practicum for computational social scientists. *arXiv Prepr arXiv220101608*.

[113] Jalal N, Ghafoor KZ. Machine Learning Algorithms for Detecting and Analyzing Social Bots Using a Novel Dataset. *ARO-THE Sci J KOYA Univ* 2022; 10: 11–21.

[114] Socialmedia Research Foundation. NodeXLgraph Gallery (#actonclimate), https://nodexlgraphgallery.org/Pages/Graph.aspx?graphID=274361 (2022).

[115] Socialmedia Research Foundation. NodeXLgraphGallery (#IStandWithPutin OR #IStandWithRussia...), https://nodexlgraphgallery.org/Pages/Graph.aspx?graphID=275885 (2022).

[116] Tharwat A. (2018). Classification assessment methods. *Appl Comput Informatics* 2018. https://doi.org/10.1016/j.aci.2018.08.003

[117] Visalakshi S, Radha V. A literature review of feature selection techniques and applications: Review of feature selection in data mining. In: *2014 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, pp. 1–6.

[118] Tardelli S, Avvenuti M, Tesconi M, et al. Characterizing social bots spreading financial disinformation. In: *International conference on human-computer interaction*. 2020, pp. 376–392.

[119] Twitter.com. Automation Rules, https://help.twitter.com/en/rules-and-policies/twitter-automation (2017, accessed 1 September 2022).

[120] Rossi E, Chamberlain B, Frasca F, et al. Temporal graph networks for deep learning on dynamic graphs. *arXiv Prepr arXiv200610637*.

[121] Tabassum F, Mubarak S, Liu L, et al. How Many Features Do We Need to Identify Bots on Twitter? In: Sserwanga I, Goulding A, Moulaison-Sandy H, et al. (eds) *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity*. Cham: Springer Nature Switzerland, 2023, pp. 312–327.

[122] Alothali E, Alashwal H, Salih M, et al. Real Time Detection of Social Bots on Twitter Using Machine Learning and Apache Kafka. In: *2021 5th Cyber Security in Networking Conference (CSNet)*. 2021, pp. 98–102.

[123] Alothali E, Hayawi K, Alashwal H. SEBD: A Stream Evolving Bot Detection Framework with Application of PAC Learning Approach to Maintain Accuracy and Confidence Levels. *Appl Sci*; 13. Epub ahead of print 2023. DOI: 10.3390/app13074443.

[124] Wu H, Shang Z, Wolter K. Trak: A testing tool for studying the reliability of data delivery in apache kafka. *Proc - 2019 IEEE 30th Int Symp Softw Reliab Eng Work ISSREW 2019* 2019; 394–397.

[125] Hamami F, Dahlan IA. The Implementation of Stream Architecture for Handling Big Data Velocity in Social Media. *J Phys Conf Ser*; 1641. Epub ahead of print 2020. DOI: 10.1088/1742-6596/1641/1/012021.

[126] Brody S, Alon U, Yahav E. How attentive are graph attention networks? *arXiv Prepr arXiv210514491*. 2021. Accessed from https://arxiv.org/abs/2105.14491. Accessed on: 30th November 2022

[127] Prechelt L. Early Stopping --- But When? In: Montavon G, Orr GB, Müller K-R (eds) *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 53–67.

[128] Ng LHX, Robertson DC, Carley KM. Stabilizing a supervised bot detection algorithm: How much data is needed for consistent predictions? *Online Soc Networks Media* 2022; 28: 100198.

[129] Pan SJ. Transfer learning. *Data Classif Algorithms Appl* 2014; 21: 537–570.

[130] Tan C, Sun F, Kong T, et al. A survey on deep transfer learning. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018, pp. 270–279.

[131] Haussler D. *Probably approximately correct learning*. University of California, Santa Cruz, Computer Research Laboratory Santa~…, 1990.

[132] Yang C, Harkreader R, Gu G. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Trans Inf Forensics Secur* 2013; 8: 1280–1293.

[133] De Nicola R, Petrocchi M, Pratelli M. On the efficacy of old features for the detection of new bots. *Inf Process Manag* 2021; 58: 102685.

[134] Gilani Z, Farahbakhsh R, Tyson G, et al. A large-scale behavioural analysis of bots and humans on twitter. *ACM Trans Web*; 13. Epub ahead of print 2019. DOI: 10.1145/3298789.

[135] Grzenda A, Kraguljac N V, McDonald WM, et al. Evaluating the Machine Learning Literature: A Primer and User's Guide for Psychiatrists. *Am J Psychiatry* 2021; 178: 715–729.

[136] Barbon S, Campos GFC, Tavares GM, et al. Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets. *ACM Trans Multimed Comput Commun Appl* 2018; 14: 26.

[137] Blaise A, Mihailescu E, Vidalenc B, et al. Learning Model Generalisation for Bot Detection. In: *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*. New York, NY, USA: Association for Computing Machinery, pp. 57–63.

[138] Feng S, Tan Z, Wan H, et al. TwiBot-22: Towards graph-based Twitter bot detection. *arXiv Prepr arXiv220604564*.

[139] Koggalahewa D, Xu Y, Foo E. An unsupervised method for social network spammer detection based on user information interests. *J Big Data* 2022; 9: 7.

List of Other Publications

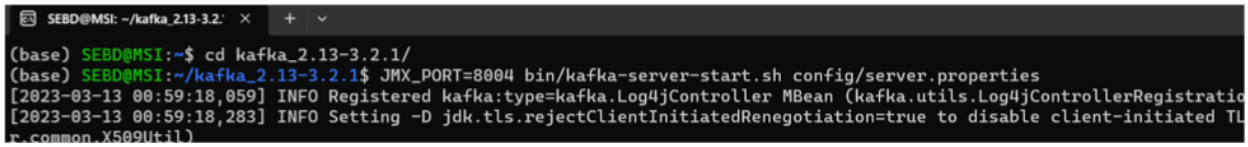
1. Alothali E, Hayawi K, Alashwal H. SEBD: A Stream Evolving Bot detection framework with application of PAC Learning Approach to maintain accuracy and confidence levels. Applied Sciences 2023. https://doi.org/10.3390/app13074443
2. Alothali E, Salih M, Hayawi K, Alashwal H. Bot-MGAT: A Transfer Learning Model Based on a Multi-View Graph Attention Network to Detect Social Bots. Applied Sciences 2022;12(16):8117. https://doi.org/10.3390/app12168117
3. Alothali E, Hayawi K, Alashwal H. Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter. Social Network Analysis and Mining 2021;11(1):1–15. https://doi.org/10.1007/s13278-021-00786-4
4. Alothali E, Alashwal H, Harous S. Data Stream Mining Techniques: A Review. TELKOMNIKA (Telecommunication Computing Electronics and Control) 2019;17(2). DOI: http://doi.org/10.12928/telkomnika.v17i2.11752
5. Alothali E, Hayawi K, Alashwal H. Using Self-labeling and Co-Training to Enhance Bots Labeling in Twitter. 2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates 2022:1–2. https://doi.org/10.1109/AICCSA56895.2022.10017585
6. Alothali E, Alashwal H, Salih M, Hayawi K. Real Time Detection of Social Bots on Twitter Using Machine Learning and Apache Kafka. 2021 5th Cyber Security in Networking Conference (CSNet), Abu Dhabi, United Arab Emirates 2021:98–102. https://doi.org/10.1109/CSNet52717.2021.9614282
7. Alothali E, Hayawi K, Alashwal H. Characteristics of Similar-Context Trending Hashtags in Twitter: A Case Study. ICWS 2020: 27th International Conference, Held as Part of the Services Conference Federation, SCF 2020, September 18–20, Honolulu, HI, USA. https://doi.org/10.1007/978-3-030-59618-7_10
8. Alothali E, Zaki N, Mohammed E, Alashwal H. Detecting Social Bots on Twitter: A Literature Review. 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates 2018:175–180. https://doi.org/10.1109/INNOVATIONS.2018.8605995
9. Alothali E, Alashwal H, Harous S. Data Stream Mining Techniques: A Review. International Conference and Workshop on Telecommunication, Computing, Electronics and Control (ICW-TELKOMNIKA) September 2018, Yogyakarta, Indonesia.

Appendix

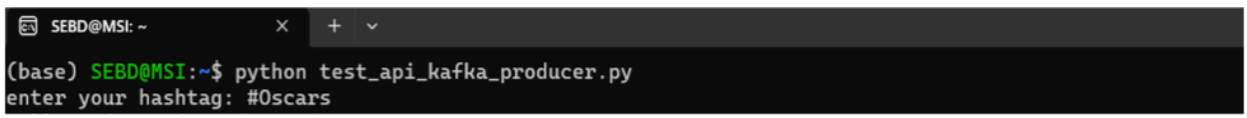
SEBD Screenshots.

1) Start ZooKeeper to store prediction

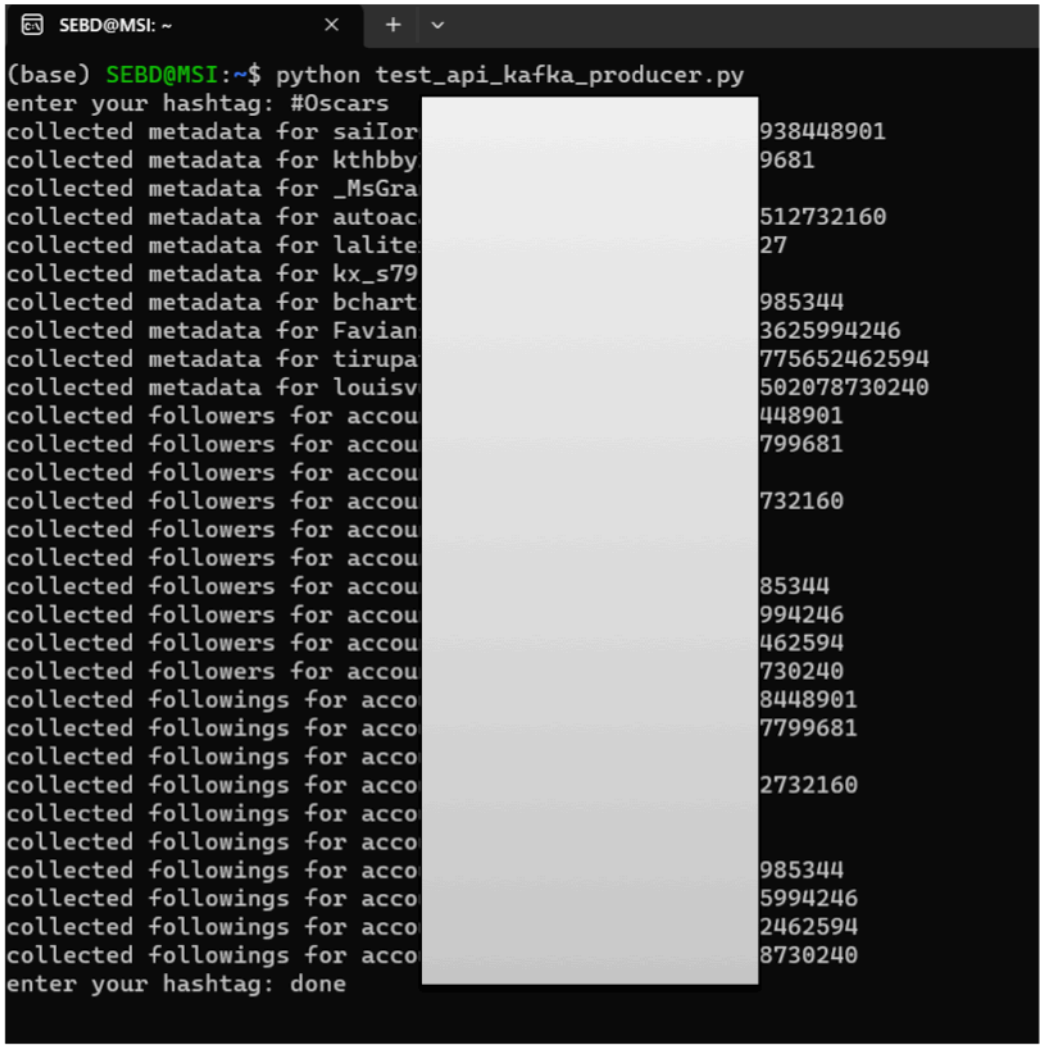
2) Start Kafka Server



3) Start Kafka producer (make sure to define the desire number of follower / friends in the code file)

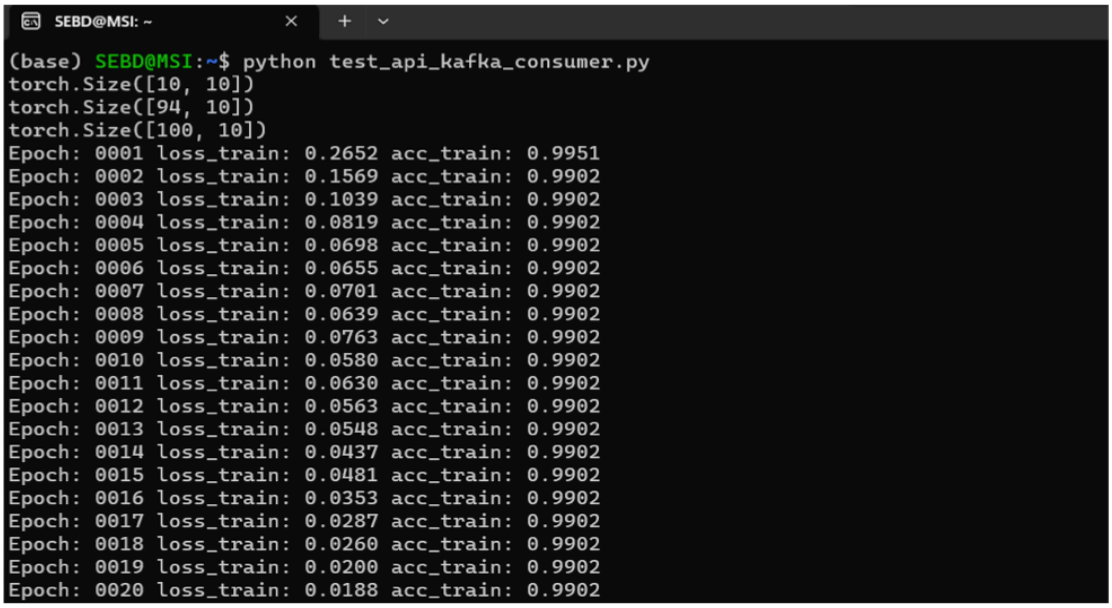


4) Enter a hashtag or keyword and wait for the constructing of the graph

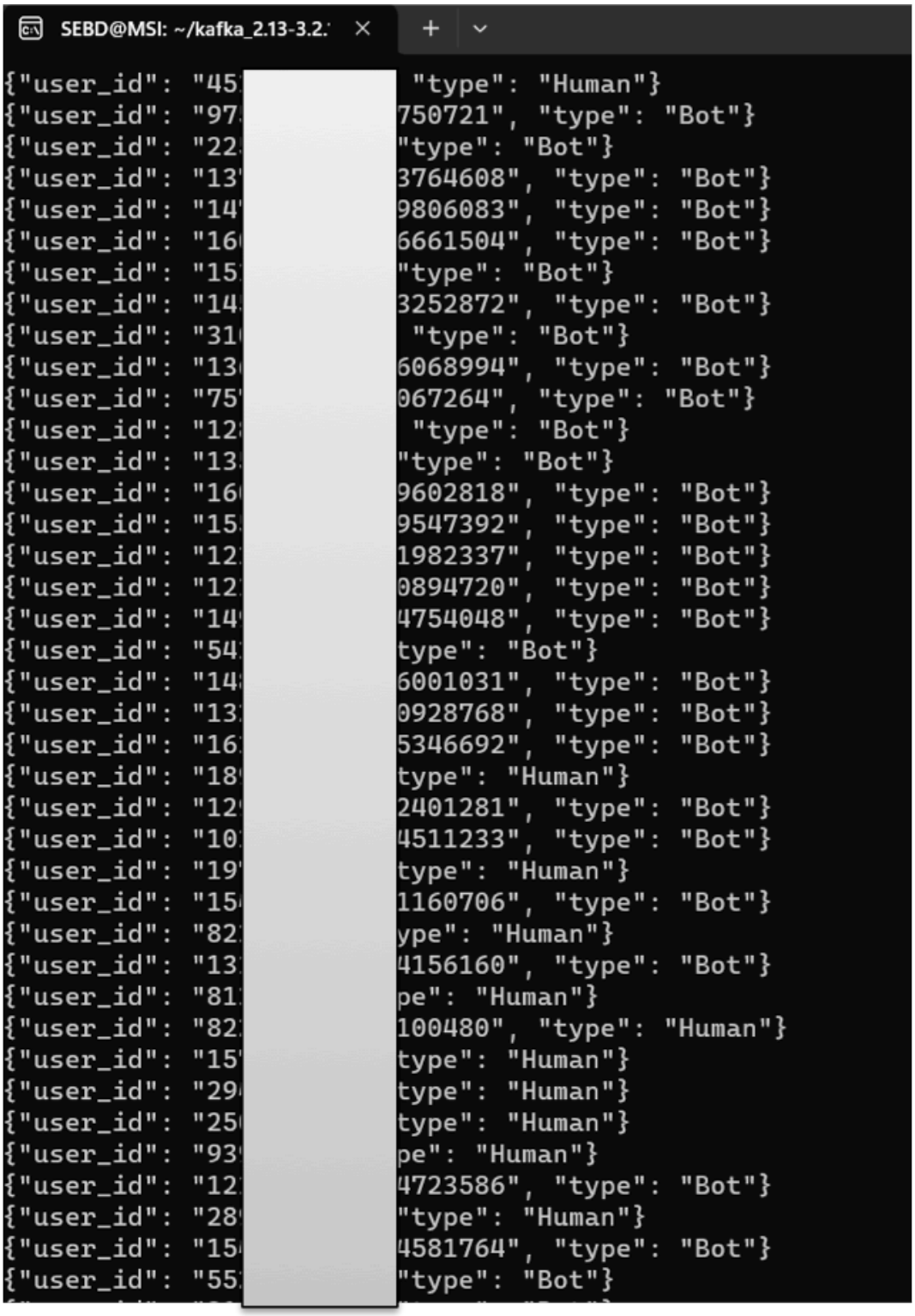


5) An option to add another hashtag to be combined with the previous one for processing at once, otherwise terminate the producer by enter word “done”

6) Start Kafka consumer to show the view of graph for users, then followers, follower of followers (x , F) where x is the number of users, F is the number of features. For example, here we have 10 users with 10 features fix for each view, then for the 10 users we retrieve the 10 followers for each user total of 94, then the same for following friends.



7) Read the prediction per user id. or open another Kafka consumer that have the saving code to csv (make sure the same topic name) to save the users with features and predicted labels.



UAEU
جامعة الإمارات العربية المتحدة
United Arab Emirates University

UAE UNIVERSITY DOCTORATE DISSERTATION NO. 2023: 31

This dissertation focuses on the problem of evolving social bots in online social networks, particularly Twitter. Such accounts spread misinformation and inflate social network content to mislead the masses. The main objective of this dissertation is to propose a stream-based evolving bot detection framework (SEBD), which was constructed using both graph- and feature-based models. Our findings indicate that Twitter has higher percentage of bots as users compared to human. Results show that stream detection is more effective than offline detection.

Eiman Alothali received her PhD from the Department of Computer Science and Software Engineering, College of Information Technology at UAE University, UAE. She received her MSc in Cyber Security from the Zayed University, UAE.

www.uaeu.ac.ae

Online publication of dissertation:
https://scholarworks.uaeu.ac.ae/etds/

UAEU عمادة المكتبات Libraries Deanship
جامعة الإمارات العربية المتحدة
United Arab Emirates University
Digital Library Services Section - قسم الخدمات المكتبية الرقمية