Authorship Attribution in the Dark Web

by

Britta Sennewald

Bachelor of Computer Science, University of Applied Sciences
Bonn-Rhein-Sieg, Germany, 2017
Bachelor of Arts, Muthesius University of Fine Arts and Design,
Germany, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s): Kenneth Kent, Ph.D., Computer Science
Rainer Herpers, Dr., Computer Science

Examining Board: Rongxing Lu, Ph.D., Computer Science
C. Paul Cook, Ph.D., Computer Science
Dhirendra Shukla, Ph.D., Technology Management and Entrepreneurship

This thesis is accepted by the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK
April, 2020

© Britta Sennewald, 2020
Abstract

This thesis is about authorship attribution (AA) within multiple Dark Web forums and the question of whether AA is possible beyond the boundaries of a single forum. AA can become a curse for users that try to protect their anonymity and simultaneously become a blessing for law enforcement groups that try to track users. To determine to what extent AA threatens the anonymity of Dark Web users, a dataset of four Dark Web forums was created. Within the analysis, two different approaches are considered: feeding classifiers with posts from two forums, and training classifiers with posts from another forum than what is used for testing. Even for the largest dataset, the author of a post is at least 94% within the top three most likely candidates. This shows that AA can be a danger to the anonymity of Dark Web users across the boundaries of different forums.
Acknowledgements

First of all I would like to thank Prof. Dr. Kenneth Kent, Prof. Dr. Rainer Herpers and Prof. Markus Ullmann for their scientific supervision of my work.

I would also like to thank Prof. Dr. Kenneth Kent, Prof. Dr. Rainer Herpers, Prof. Dr. Karl Jonas, Prof. Dr. Patricia Evans, Nadine Kutz, Jana Schiewe and Jodi O’Neill, who actively supported me on my way to the dual degree.

Many thanks to Stefan Donath that you were always there for me.

Many thanks to my parents Ulla and Jörg Sennewald for always encouraging me.

A big thank you also to Angela Rosslenbroich, Christina Zimmermann, Hannah Weger, Gabi Donath, Ulf Donath, Simon Donath, Sabine Döring, Anna Afijan, Janna Derwahl and Dr. Ida S.R. Siahaan for supporting me at all times, no matter how stressed I was.
# Table of Contents

**Abstract**  

**Acknowledgments**  

**Table of Contents**  

**List of Tables**  

**List of Figures**  

**List of Symbols, Nomenclature or Abbreviations**  

## 1 Introduction  

## 2 Related Work and Theoretical Background  

---  

### 2.1 Related Work  

### 2.2 Authorship Attribution  

#### 2.2.1 Number of Authors  

#### 2.2.2 Features for AA  

#### 2.2.3 Techniques for AA  

### 2.3 Dark Web  

#### 2.3.1 The Onion Router (TOR)  

#### 2.3.1.1 The Basic Principle of TOR  

#### 2.3.1.2 Onion Service  

#### 2.3.2 Pretty Good Privacy  

---
3 Artificial Intelligence

3.1 Natural Language Processing ........................................... 18

3.1.1 Basics of NLP ......................................................... 18

3.1.1.1 Linguistics ....................................................... 19

3.1.1.2 Syntax and Structure .......................................... 19

3.1.2 NLP Techniques ..................................................... 21

3.1.2.1 Preprocessing ................................................... 21

3.1.2.2 Feature Engineering ............................................ 24

3.1.2.3 Topic Modelling ................................................ 27

3.2 Machine Learning ....................................................... 29

3.2.1 Basics of ML ......................................................... 29

3.2.1.1 Categories of Variables ....................................... 29

3.2.1.2 Selection of an Algorithm ................................... 30

3.2.1.3 Evaluation of the Results .................................... 31

3.2.2 Support Vector Machine ........................................... 33

3.2.2.1 Fundamental Functionality of two-class Support Vec-
ton Machines ......................................................... 33

3.2.2.2 Hard-Margin Support Vector Machines ..................... 34

3.2.2.3 Fundamental Functionality of multi-class Support Vec-
ton Machines ......................................................... 37

3.2.3 Decision Trees ....................................................... 38

3.2.3.1 Information Content of Features ........................... 39

3.2.4 Naive Bayes ......................................................... 41

3.2.4.1 Probabilities .................................................... 41

3.2.4.2 Discrete Attributes ........................................... 42

4 Dataset ........................................................................ 44

4.1 Creating a new Dataset ................................................ 44
List of Tables

4.1 Statistics of all four crawled Dark Web forums, showing an overview of the dataset used in this thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

5.1 Remaining forum combinations and number of corresponding authors after filtering out all authors with less than 50 posts in both forums. . 64

C.1 List of all features used within this analysis. . . . . . . . . . . . . . . 121

C.2 List of all POS tags considered within the stylometric features. . . . 122

D.1 Listing of all 24 analyses of type I . . . . . . . . . . . . . . . . . . . . . 124

D.2 Listing of all 28 analyses of type II . . . . . . . . . . . . . . . . . . . . . 125

D.3 Absolute frequencies with which the listed classifiers achieved the best result for the respective feature category in the 52 analyses of this thesis.128
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Number of authors considered in different AA research papers</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Features considered in different AA research papers</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Techniques considered in different AA research papers</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>The proportions of Surface Web, Deep Web and Dark Web</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Visualization of the Tor circuit established between a client (Alice) and a public server within the surface web.</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Visualization of the connection between a client (Alice) and an anonymous server (hidden service).</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Many words without order or relation</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Words are ordered and structured by applying the hierarchical syntax</td>
<td>20</td>
</tr>
<tr>
<td>3.3</td>
<td>An example of a word stem and its inflections</td>
<td>23</td>
</tr>
<tr>
<td>3.4</td>
<td>Basic principle of the Latent Dirichlet Allocation</td>
<td>28</td>
</tr>
<tr>
<td>3.5</td>
<td>The basic principle of cross-validation</td>
<td>33</td>
</tr>
<tr>
<td>3.6</td>
<td>Difference between hard-margin and soft-margin SVMs</td>
<td>34</td>
</tr>
<tr>
<td>3.7</td>
<td>The hyperplane with a maximum distance between two classes</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Screenshot of the start page of the Dark Web forum DNM Avengers</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Screenshot of the start page of the Dark Web forum The Hub</td>
<td>52</td>
</tr>
<tr>
<td>4.3</td>
<td>Screenshot of the start page of the Dark Web forum The Majestic Garden</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Screenshot of the start page of the Dark Web forum Dread</td>
<td>54</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.1</td>
<td>Visualisation of analysis type I and analysis type II</td>
<td>68</td>
</tr>
<tr>
<td>5.2</td>
<td>Sum of all posts within the corresponding forum combination and type of analysis</td>
<td>69</td>
</tr>
<tr>
<td>5.3</td>
<td>Unbalanced distribution of the number of post per author within each forum combination/type of analysis</td>
<td>70</td>
</tr>
<tr>
<td>6.1</td>
<td>Overview of the F1-Score achieved within the analyses of posts from all authors of all six forum combinations (analysis type I)</td>
<td>73</td>
</tr>
<tr>
<td>6.2</td>
<td>Overview of the F1-Score achieved within the analyses of posts from all authors of all six forum combinations (analysis type II)</td>
<td>74</td>
</tr>
<tr>
<td>6.3</td>
<td>Probability to find the correct author of a post within the top three when considering time-based features only and all authors (FC-5 and FC-6)</td>
<td>75</td>
</tr>
<tr>
<td>6.4</td>
<td>Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type I and time-based features only</td>
<td>77</td>
</tr>
<tr>
<td>6.5</td>
<td>The activity within the day of two Dark Web forum users</td>
<td>78</td>
</tr>
<tr>
<td>6.6</td>
<td>The daily activity of all users that are considered within FC-6</td>
<td>79</td>
</tr>
<tr>
<td>6.7</td>
<td>Probability to find the correct author of a post within the top three when considering stylometric-based features only and all authors (FC-5 and FC-6)</td>
<td>81</td>
</tr>
<tr>
<td>6.8</td>
<td>Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type I and stylometric-based features only</td>
<td>83</td>
</tr>
<tr>
<td>6.9</td>
<td>Probability to find the correct author of a post within the top three when considering lexical-based features only and all authors (FC-5 and FC-6)</td>
<td>86</td>
</tr>
<tr>
<td>6.10</td>
<td>Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type I and lexical-based features only</td>
<td>88</td>
</tr>
</tbody>
</table>
6.11 Probability to find the correct author of a post within the top three when considering language model-based features only and all authors (FC-5 and FC-6) ................................................................. 90

6.12 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type I and language model-based features only ................................................................. 91

6.13 Probability to find the correct author of a post within the top three when considering all features combined and all authors (FC-5 and FC-6) ............................... 93

6.14 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type I and all features combined ................................................................. 94

6.15 Probability to find the correct author of a post within the top three when voting all features and consider all authors (FC-5 and FC-6) .................................................................................................................. 96

6.16 Detailed overview of all results (incl. subfeatures) achieved by analysis type I (balanced) of FC-5 and FC-6 ................................................................................................................. 98

6.17 Detailed overview of all results (incl. subfeatures) achieved by analysis type II (balanced) of FC-5 and FC-6 ................................................................................................................. 99

A.1 Threshold ($\theta_i$) computation method to convert continues values into boolean values ............................................................................................................................... 117

A.2 Discretization of continuous attributes by using intervalls (top) or a function $p(x)$ (bottom) ................................................................. 118

A.3 Visualization of a gaussian-based pdf ................................................................................................................................. 118

B.1 Captchas of Dread ......................................................................................................................................... 119

B.2 Unsolved captcha from Dread ......................................................................................................................................... 119

B.3 Solved captcha from Dread ......................................................................................................................................... 119
D.1 Overview of the F1-Score achieved within the analyses of posts from authors with more than 500 or more posts of all remaining three forum combinations (analysis type I) ............................................. 126

D.2 Overview of the F1-Score achieved within the analyses of posts from authors with more than 1000 or more posts of all remaining three forum combinations (analysis type I) ............................................. 126

D.3 Overview of the F1-Score achieved within the analyses of posts from authors with more than 200 or more posts of all remaining five forum combinations (analysis type II) ............................................. 127

D.4 Overview of the F1-Score achieved within the analyses of posts from authors with more than 400 or more posts of all remaining three forum combinations (analysis type II) ............................................. 127

D.5 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type II and time-based features only ............................................. 128

D.6 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type II and stylometric features only ............................................. 129

D.7 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type II and lexical-based features only ............................................. 129

D.8 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type II and language model-based features only ............................................. 130

D.9 Detailed results of all authors from FC-5 and FC-6 when considering all authors within analysis type II and all features combined ............................................. 130

D.10 Statistics on the post lengths of each author from FC-5 (analysis type I) ........................................................................................................... 131

D.11 Statistics on the post lengths of each author from FC-6 (analysis type I) ........................................................................................................... 132

D.12 Statistics on the post lengths of each author from FC-5 (analysis type II, unbalanced) ........................................................................................................... 133
| D.13 | Statistics on the post lengths of each author from FC-5 (analysis type II, balanced) | 134 |
| D.14 | Statistics on the post lengths of each author from FC-6 (analysis type II, unbalanced) | 135 |
| D.15 | Statistics on the post lengths of each author from FC-6 (analysis type II, balanced) | 136 |
| D.16 | Confusion matrix of analysis type I, FC-5, all authors, unbalanced | 137 |
| D.17 | Confusion matrix of analysis type I, FC-5, all authors, balanced | 138 |
| D.18 | Confusion matrix of analysis type I, FC-6, all authors, unbalanced | 139 |
| D.19 | Confusion matrix of analysis type I, FC-6, all authors, balanced | 140 |
| D.20 | Confusion matrix of analysis type II, FC-5, all authors, unbalanced | 141 |
| D.21 | Confusion matrix of analysis type II, FC-5, all authors, balanced | 142 |
| D.22 | Confusion matrix of analysis type II, FC-6, all authors, unbalanced | 143 |
| D.23 | Confusion matrix of analysis type II, FC-6, all authors, balanced | 144 |
| D.24 | Confusion matrix of analysis type I, FC-5, all authors, voting, unbalanced | 145 |
| D.25 | Confusion matrix of analysis type I, FC-5, all authors, voting, balanced | 146 |
| D.26 | Confusion matrix of analysis type I, FC-6, all authors, voting, unbalanced | 147 |
| D.27 | Confusion matrix of analysis type I, FC-6, all authors, voting, balanced | 148 |
| D.28 | Confusion matrix of analysis type II, FC-5, all authors, voting, unbalanced | 149 |
| D.29 | Confusion matrix of analysis type II, FC-5, all authors, voting, balanced | 150 |
| D.30 | Confusion matrix of analysis type II, FC-6, all authors, voting, unbalanced | 151 |
| D.31 | Confusion matrix of analysis type II, FC-6, all authors, voting, balanced | 152 |
D.32 Detailed overview of all results (incl. subfeatures) achieved by analysis type I (unbalanced) of FC-5 and FC-6 ............................................. 153
D.33 Detailed overview of all results (incl. subfeatures) achieved by analysis type II (unbalanced) of FC-5 and FC-6 ............................................. 153
# List of Symbols, Nomenclature or Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Authorship Attribution</td>
</tr>
<tr>
<td>AC</td>
<td>Authorship Characterization</td>
</tr>
<tr>
<td>AD</td>
<td>Authorship Detection</td>
</tr>
<tr>
<td>ADI</td>
<td>Authorship De-Identification</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>AV</td>
<td>Authorship Verification</td>
</tr>
<tr>
<td>BOW</td>
<td>Bag-Of-Words</td>
</tr>
<tr>
<td>CBOW</td>
<td>Continuous Bag of Words</td>
</tr>
<tr>
<td>CnG</td>
<td>Common N-Grams</td>
</tr>
<tr>
<td>CS</td>
<td>Cosine Similarity</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
</tr>
<tr>
<td>DF</td>
<td>Decision Functions</td>
</tr>
<tr>
<td>DNMA</td>
<td>DNM Avengers</td>
</tr>
<tr>
<td>DT</td>
<td>Decision Tree</td>
</tr>
<tr>
<td>ECOC</td>
<td>Error-Correcting Output Code</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>GloVe</td>
<td>Global Vectors for Word Representation</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>HS</td>
<td>Hidden Services</td>
</tr>
<tr>
<td>I2P</td>
<td>Invisible Internet Project</td>
</tr>
<tr>
<td>IDF</td>
<td>Inverse Document Frequency</td>
</tr>
<tr>
<td>IGN</td>
<td>Information Gain</td>
</tr>
<tr>
<td>k-NN</td>
<td>K-Nearest Neighbour</td>
</tr>
<tr>
<td>KLD</td>
<td>Kullback-Leibler Divergence</td>
</tr>
<tr>
<td>LDA</td>
<td>Latent Dirichlet Allocation</td>
</tr>
<tr>
<td>LSA</td>
<td>Latent Semantic Analysis</td>
</tr>
<tr>
<td>MACs</td>
<td>Message Authentication Codes</td>
</tr>
<tr>
<td>MC</td>
<td>Markov Chains</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi Layer Perceptrons</td>
</tr>
<tr>
<td>NB</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NMF</td>
<td>Non-Negative Matrix Factorization</td>
</tr>
<tr>
<td>OS</td>
<td>Onion Services</td>
</tr>
<tr>
<td>OTS</td>
<td>One-Time-Secret</td>
</tr>
<tr>
<td>PD</td>
<td>Plagiarism Detection</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PGP</td>
<td>Pretty Good Privacy</td>
</tr>
<tr>
<td>POS</td>
<td>Part of Speech</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forest</td>
</tr>
<tr>
<td>RLP</td>
<td>Recentred Local Profile</td>
</tr>
<tr>
<td>RP</td>
<td>Rendezvous Point</td>
</tr>
<tr>
<td>SB</td>
<td>Social Behavioral</td>
</tr>
<tr>
<td>SCAP</td>
<td>Source Code Author Profile</td>
</tr>
<tr>
<td>SMO</td>
<td>Sequential Minimal Optimization</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TF</td>
<td>Term Frequency</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>Term-Frequency Inverse Document-Frequency</td>
</tr>
<tr>
<td>TH</td>
<td>The Hub</td>
</tr>
<tr>
<td>TIS</td>
<td>Text Indexing and Segmentation</td>
</tr>
<tr>
<td>TMG</td>
<td>The Majestic Garden</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TOR</td>
<td>The Onion Router</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>VS</td>
<td>Vector Space</td>
</tr>
<tr>
<td>WOT</td>
<td>Web-of-Trust</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Dark Web technologies such as The Onion Router (TOR) are getting increasingly popular \[43\]. They offer the opportunity to access the internet anonymously via multi-layer encrypted connections. This is interesting especially for users who want to protect their privacy, circumvent censorship on the internet, or for those who do not want to be identified when performing illegal activities \[69\]. The Dark Web provides anonymity for everybody. Due to technologies like TOR or the Invisible Internet Project (I2P), it is a difficult task for law enforcement groups to trace back the track of criminal users within the Dark Web. By using multiple accounts with different pseudonyms within different Dark Web marketplaces and forums, normal users, as well as criminals, try to split their activity (e.g., malicious and benign activities in the case of criminals) to keep their (cyber)-identity secret from others \[69\].

Authorship Attribution (AA) focuses on assigning documents to their corresponding authors. This is very successful when a sufficient amount of text is available. If the length of a text and/or the number of texts is small, it becomes increasingly challenging to attribute them correctly to an author \[71\]. Transferred to the application domain Dark Web, this implies, that posts within Dark Web forums could be assigned to their authors via authorship attribution. The main difficulty for AA in this
domain is the limited number of posts that are available for each candidate author as well as comparatively short texts. Some researchers have already concentrated on identifying users within the Dark Web, but in most cases, the focus was on solving this problem by analyzing the network traffic of TOR, I2P or other anonymization techniques (see e.g., Johnson et al. [35] or Nurmi et al. [53]). Nonetheless, in a very few studies, the focus is on a combination of AA within a Dark Web forum (e.g., Ho and Ng [29] or Spitters et al. [69]). Compared to these studies, this thesis focuses on AA within multiple forums and not only within a single one. In addition, instead of using a single classifier to analyze the data, a voting approach based on multiple Machine Learning (ML) classifiers is used in this thesis to attribute posts to their authors.

Applying AA to Dark Web forums can have positive as well as negative facets. A good AA algorithm is of interest to law enforcement agencies who want to track users engaged in illegal activities. However, the same algorithm can be used to identify users who are dependent on the anonymity of the Dark Web to be able to express their opinions freely and thus avoid the suppression of regimes. Therefore AA can be both, a way to track criminals, as well as a danger to privacy on the Dark Web. For these two reasons, it is interesting to investigate how precise posts can be assigned to their authors on the Dark Web especially between multiple forums.

In this thesis, authorship attribution will be applied to posts published by authors that are active in two different Dark Web forums. For this purpose, different techniques like Natural Language Processing (NLP) and ML are used. By doing so, it is possible to figure out to which extent posts can be attributed to their correct author regardless of which of the two forums the posts were published or which username was used. The dataset needed for this analysis will be created by crawling/scraping four different Dark Web forums.
2.1 Related Work

Authorship attribution is a difficult task in an environment with hundreds or even thousands of different authors and texts that differ greatly in size. Many researchers already focused on authorship attribution and also on the Dark Web. Nonetheless, only a few researchers have concentrated on a combination of both. This combination is especially interesting, because it allows researchers to determine to what extent the anonymity of Dark Web users is endangered by AA and ML.

Ho and Ng analyzed stylometric features of texts posted in different Dark Web forums in [29]. They tried to connect ten authors within Dark Web forums by extracting lexical features and then classify them with a Support Vector Machine (SVM) classifier. Each user wrote at least 400 posts and around 6000 words in the English language. For validation, they split the post from one author up into two parts which were denoted as the positive training and testing sets, respectively. As a negative training set, they used posts of 99 other authors. Hence, they trained a
classifier for each author, which means that they trained and tested their classifier with the posts of 100 authors in total.

Linking users within different Dark Web forums is one of the main tasks in this thesis and is quite challenging. Fortunately, even when not in the context of authorship attribution, Me, Spagnoletti and Pesticcio focused on the relationship between different TOR marketplace users by concentrating on their Pretty Good Privacy (PGP) keys [46]. PGP-keys are based on the concept of Web-of-Trust (WOT) and are used (in the case of a Dark Web marketplace) to provide confidentiality within a transaction. Hence, the public key of a user (especially those from vendors and buyers) can be found in their user profile. The researchers were able to extract relations of a key owner to those users who signed the key of the owner by using the WOT structure. Even when the original research problem of this paper does not belong to authorship attribution, the information that is important at this point is, that Dark Web users are using PGP keys for their transactions and therefore can be linked by these keys. However, not all users will have just one key for all Dark Web forums they are active in, but those who do use a key in two forums can be linked properly.

Having said that, another research project regarding Dark Web and authorship attribution is undertaken by Spitters et al. The authors were able to get an accuracy of around 97% to list the actual author of a post in the top five most highly ranked candidates [69]. They achieved their results by using a combination of time-based and stylometric features as well as character trigrams. In contrast to Ho and Ng [29], the authors of this paper transferred the AA problem into a conventional multi-class classification task. For the classification, they also used Support Vector Machines. Furthermore, the authors first deleted all uninformative posts before they combined n of them in an instance to get a more robust representation of an author’s writing. The authors also analyzed the performance of the classifier when it is trained with the features separately or with a combination of them. By using all three features
together, the actual author was ranked at the first rank with a probability of 88% and ranked within the top five most highly ranked candidates with a probability of 97%.

Ashcroft et al. tried to match multiple aliases of the same user within a data set of an Irish web forum and within a Twitter dataset [16]. Even when this research does not focus on the Dark Web environment, it tries to match users within different domains by using authorship attribution techniques. The authors used stylometric and time-based features like other researchers, but they also added so-called emotion-based or Twitter-specific features to their analysis. These features contain emotion words, smileys, hashtags, URLs and how frequently other users are mentioned within the posts. Similar to Spitters et al. the authors analyzed the performance of their approach by classifying the post of a user by using those three features separately as well as in combination. For doing so the authors used three different machine learning classifiers AdaBoost, Support Vector Machine, and Naive Bayes (NB). They achieved the highest accuracy by using the combination of all three features and the Ada Boost classifier. In the last step, the authors applied their method to 4500 bloggers. Some of them authored multiple blogs and could be linked by comparing their Google account. The approach was able to achieve a precision of 96.6% and a recall of 57%.

The scientists Pillay and Solorio also worked on AA of Web Forum posts [60]. They used stylometry features (lexical and syntactical), statistical language models, clustering and different machine learning algorithms in their work. Clustering was applied to use the output as meta-features for identifying the authors. For analyzing the posts, they combined all writings of an author into one single text. For the classification, the authors used three different classifiers: BayesNet, Naive Bayes, and Decision Trees (C4.5). The results show, that their approach is able to classify posts of five authors with a probability of around 90% correctly by using BayesNet. How-
ever, by an increasing number of authors, C4.5 seems to be the better choice than BayesNet. Having said this, the probability to identify the correct author within a group of 10, 15, 30, 50, or 100 candidates drops to an accuracy of 74% when considering 30 authors or lower. Interestingly, the accuracy achieved when analyzing 10 authors is only 64%, which means that the dataset could not be classified as good as the dataset with 30 authors. The same applies to the dataset with 15 authors that achieved an accuracy of 72%. A similar tendency is visible when focusing on the results of the analyses of the datasets with 100 authors and 50 authors. In the former case, the accuracy is 2% higher even when the number of candidate authors is doubled. Besides that, the authors observed that when the number of authors increases, those classifiers that incorporate a cluster identifier worked best.

M. Sultana, P. Polash, and M. Gavrilova have focused on an extreme version of the problem AA methods have to face nowadays [70]: extremely short texts like those on Twitter have become increasingly popular. Other characteristics are that they are written in an informal way, the topic varies over time, and they contain a rich mix of weblinks, hashtags, emojis and so on. Hence, applying traditional AA on this kind of text might become quite difficult. The authors of this paper tried to overcome the problem by building Social Behavioral (SB) profiles of each user. These profiles are developed by analyzing social interactive data like replied friends, shared hashtags and retweets for their analysis. To compare their approach with traditional methods, they also created linguistic profiles by using Bag-Of-Words (BOW) and Style-Markers. However, they could achieve a recognition rate above 95% for 70 analyzed twitter users by using SB profiles. Furthermore, the authors could show, that the recognition rate of style markers is better than the one of BOW. With that in mind, not all of the posts posted in dark web forums are as long as needed for traditional AA methods. Hence, analyzing the social behavior of dark web users might be an interesting feature.
As mentioned before, forum posts also provide another interesting feature for AA, the time when a post was posted. Even when the research of La Morgia et al. [50] is not in the context of authorship attribution, it shows that only the activity of a user can reveal his/her location. The authors analysed the activity of users of five different dark web forums. In the case of two of them, they already knew where most of the users are coming from. Hence, their goal was to figure out where the users of the other three forums were coming from. To get a ground truth about how the activity of users looks around the world, they used a Twitter data set with known origin of all users. After that, they were able to distinguish groups of dark web users from different regions around the globe within a crowd just by analysing the timestamps of posts. Therefore it can be concluded, that the activity of users that are living in a different timezone is different regarding the time. Hence, this feature might be especially useful when analysing forums with a global community.

Swain, Mishra, and Sindhu give an overview of recent approaches on AA techniques in their paper [71]. They state that authorship analysis can be split into several different categories: authorship attribution which is about identifying an author of a set of different texts; Authorship Characterization (AC) focuses on identifying sociolinguistic attributes of an author like gender, age or educational level; Authorship Verification (AV) is about checking whether an author wrote a text or a set of similar texts or not; Authorship Detection (AD) is about figuring out if two different texts are written by the same author or not. Other categories are Plagiarism Detection (PD) as well as Text Indexing and Segmentation (TIS), which focus on differentiating segments that are written from different authors within a single, combined document, or Authorship De-Identification (ADI), which is about identifying appropriate features to detect the writing style of an author. However, after this definition part, they list multiple research projects focusing on this area, especially the category, language, domain, features, and techniques that have been used. This survey
Fig. 2.1. Number of authors considered in different AA research papers according to Swain et al. [71]. The bar chart shows only a view of papers where more than 50 authors are considered in parallel. The median number of authors according to Swain et al. [71] is 13.

shows that Naive Bayes an Support Vector Machine is the most commonly used classifier, English the most frequently analyzed language and lexical and syntactic features the most popular features.

2.2 Authorship Attribution

The main task of authorship attribution is to link documents to their authors by analyzing different features extracted from these documents [71]. In this thesis, authorship attribution is applied based on Dark Web forum posts from different authors. Since the forums are the core element of the entire thesis, it is important to determine which requirements they have to meet for AA. The same applies to the features extracted from the forum posts and the classifiers used in the final analysis. In the following sections, some statistics about previous AA related work are presented, that can be used as a guideline for this thesis and future work in the area of AA.
2.2.1 Number of Authors

In Fig. 2.1, the number of authors considered in 22 different authorship attribution research papers, extracted from the paper written by Swain, Mishra, and Sindhu is visualized. The maximum number of authors analyzed in parallel was 177 by Martijn Spitters et al. [69], whereas A. O. Kusakci analyzed only two authors [40], which is the smallest number of authors used in a paper presented by Swain et al. The average number of authors according to Fig. 2.1 is 37 authors within a single analysis. When focusing at the median, the number drops to only 13 authors that are considered in parallel within an AA research project.

However, focusing on the number of authors alone is only half of the story. The context in which AA is applied, should also play a role in this section. Even when this thesis focuses on forum posts, AA can also be applied to books, reviews, SMSs, [Image: Features used for AA]

Fig. 2.2. Features considered in different AA research papers according to [71]. It becomes obvious that approx. four different features appear within nearly every analysis, whereas all other features listed in the bar chart are used comparatively rarely.
instant messages, documents, and any other text document. The number of authors considered in parallel differs from domain to domain. The three papers with the highest number of authors were focusing on: posts within a single Dark Web forum (177 authors) [69], e-mails (158 authors) [52], and novels from Project Gutenberg (136 authors) [54]. However, only 20 to 62 authors are analyzed in papers that mainly focused on newspaper articles, general literature, online reviews, and poetry but also on instant messages and spam e-mails. Scientists that used 13 authors or fewer mainly focused on ancient texts, novels, research papers, translated texts, and multi-author documents. Spitters et al. [69] applied authorship attribution to a single Dark Web forum and used a maximum of 177 users for the analysis. When taking a deeper look into this paper, it becomes clear that they have split the dataset up into groups of 25, 50, 100 and 177 users. However, they achieved the best results when applying AA to only 25 authors, which is a little bit more than used by most of the papers (median of 13).

2.2.2 Features for AA

Various features can be extracted from texts and used for authorship attribution. Hence, Fig. 2.2 shows the features used within the papers presented by Swain et al. [71], as well as the frequency that these features are used. According to the data provided, a paper that focuses on AA uses three features on average, but at least one and a maximum of six different features at once. The chart below shows that lexical and syntactic features are most commonly used, followed by Common N-Grams (CnG) and structural features. When a text is considered as a sequence of elements or tokens that are grouped into consecutive sentences, then this belongs to the lexical feature category. Syntactic features denote specific patterns that are used to build sentences, whereas structural features focus on the way an author structures or organizes its text. CnG features do not consider all the n-grams of a text. They
Fig. 2.3. Techniques considered in different AA research papers according to [71]. The bar chart shows that in most cases a Support Vector Machine classifier is used. All other classifiers are used considerably less frequently in the papers listed in [71]. An explanation of the abbreviations on the X-axis can be found in the text below.

focus only on differences within the usage frequencies of those character n-grams, that are most common.

However, except of these top four most used features, Part of Speech (POS) tags seem to be frequently used as well as content-specific features, rare words, Term-Frequency Inverse Document-Frequency (TF-IDF) and stop words (see Section 3.1.2). Interestingly, POS tags, TF-IDF and stop words are listed separately to the lexical features by Swain et al. [71], even when the definition of lexical features of this paper might include all of them. However, this graph shows that most scientists that focus on AA use the same, well-known, old, and trusted features, instead of experimenting with new ones. As well-known and new features will be used separately and combined within this work, it will become interesting to discover which of them works best.
2.2.3 Techniques for AA

In the previous section, the chart Fig. 2.2 showed a clear trend that three to four features are used together in nearly all AA analyses, whereas the remaining features are used within just a single paper. This trend continues in this section that focuses on techniques that are used to analyze the features mentioned above. Interestingly, this time it is only one single classifier that is used nearly in all AA papers that are mentioned by Swain et al. This is the Support Vector Machine classifier whose basic functionality is explained in Section 3.2.2. Furthermore, Decision Tree (DT) classifiers and Random Forest (RF) classifiers are used ten times in papers considered by Swain et al. Another classifier that seems to be promising according to Fig. 2.3 is the Naive Bayes classifier. Except that, Multi Layer Perceptrons (MLP) or other neural networks (denoted also as MLP in the chart) seem to be worth a try as well. The same applies to Decision Functions (DF) that are used as frequently as the naive Bayes classifier.

It should be noted that in most papers three classifiers are used either together or in parallel. The minimum number of classifiers used was one whereas the maximum was six classifiers that are used within a single paper. For the sake of completeness, the complete designations of the abbreviations used within Fig. 2.3 that are not yet mentioned in the text, are Sequential Minimal Optimization (SMO) [33], K-Nearest Neighbour (k-NN) [39], Source Code Author Profile (SCAP) [27], Markov Chains (MC) [33], Cosine Similarity (CS) [64], Recentred Local Profile (RLP) [42], Information Gain (IGN) [39] and Kullback-Leibler Divergence (KLD) [33].

2.3 Dark Web

The well-known internet can be divided into three main subparts. These are the Surface Web, the Deep Web, and the Dark Web. The terms Dark Web and Deep
Fig. 2.4. The proportions of Surface Web, Deep Web and Dark Web visualized by using the example of an iceberg.

Web are often mixed up even when they denote two different things, which is why this subsection explains the difference between all of the three terms first before it addresses technical details.

- **Surface Web**: the Surface Web is the *normal* part of the internet that everybody knows. According to [22], its traffic is only about 10% of the total internet traffic, other sources even say that this part is only about 4%. It differs from the rest of the Internet in that its web pages are indexable. This means that they can be found with search engines like Google, Bing, etc.. Hence, it is only the tip of the iceberg as visualized in Fig. 2.4.

- **Deep Web**: the Deep Web, contains the remaining 90% or 96% of the Internet. This part of the internet contains all websites that are not indexable, and therefore can not be found by well-known search engines [30].

- **Dark Web**: the proportion of the internet that belongs to the Dark Web is only about 6%. It is part of the Deep Web but it differs from it in one essential point, namely the way a user can gain access to it. To get access to the Dark Web a special technology is needed like the TOR browser [30].
2.3.1 The Onion Router (TOR)

As already mentioned, the Dark Web can only be accessed by using special anonymization techniques. A famous technique is TOR, which will also be used in this thesis and therefore described in more detail in the following sections. However, there are also others like the I2p [7], Freenet [6], or Riffle [41].

The history of TOR can be traced back to Paul Syverson, David Goldschlag and Michael Reed from the US Naval Research Laboratory. They presented their idea in 1997 [72], which is strongly influenced by a publication of Chaum [21] from 1981. The intention behind their development was to enable anonymous surfing as well as sending emails anonymously. Their product was initially published as an open-source version in 2003. In 2006, the nonprofit organization The Onion Router was founded, which still provides the possibility to download the TOR browser [25]. Public servers like google.com are accessed in a different way than anonymous servers within the TOR network, so-called Onion Services (OS) or Hidden Services (HS). Thus, the following first subsection focuses on how a user can connect to a public server via TOR, and the second focuses on details of a connection to an OS. The latter is of particular interest because a Dark Web Forum is an OS.

2.3.1.1 The Basic Principle of TOR

Let Alice be the user who wants to get access to a public server via TOR. Assuming she uses a TOR browser, then this browser will establish a secure connection to a so-called entry guard. TOR consists of multiple relays, which are servers, that were set up and configured by volunteers. An entry guard is one of these relays. However, a relay can be placed all over the world but its IP address is always stored in a so-called directory authority whose IP, in turn, is hardcoded in the TOR browser [25]. At this point, Alice is connected to a random entry guard within the TOR network. This is neither the server she wanted to access nor provides this connection
any anonymity. Hence, further steps are necessary for a secure and anonymous connection over TOR. Therefore, Alice needs to send a request to the entry guard to establish another secured connection to a second, randomly chosen so-called middle relay within TOR. Therefore, the middle relay only knows the IP address of the entry guard [25] [45] but not the IP address of Alice. However, to be able to connect to a public server, Alice has to send another message to the entry guard that contains a request to the middle relay to establish another connection to a third server within the TOR network, the exit relay. After this last connection is set up, Alice can send a triple encrypted message (one layer of encryption per relay) to the exit relay, to establish a connection from that point on to the final public server. Thus, the last node knows the IP address of the public server and the IP address of the middle relay but neither the IP address of the entry guard nor the one of Alice [25] [45]. The resulting so-called TOR circuit with three relays is visualized in Fig. 2.5.
2.3.1.2 Onion Service

In the previous case, only one side (Alice) was anonymous whereas the IP address of the server was publicly known. In contrast, onion services like Dark Web forums and marketplaces stay anonymous as well. How this works is described below.

In the very first step, an OS needs to establish three TOR circuits to randomly chose relays, so-called introduction points, in the same manner as described in the previous section and send them its public key. After that, the service is connected to the TOR network, but nobody can access it. To solve this problem, a so-called OS descriptor is needed, that contains the service’s public key, a summary of all three introduction points, and that is signed with the private key of the OS. This descriptor has to be uploaded to a database that contains a distributed hash table for onion services. This table can be accessed by users when they are aware of the *.onion address of the service. The prefix of the address needs to be 16 characters long (version2) and has to be derived from the public key of the OS.

After the OS is successfully set up, it is now the client’s turn to connect to it. For being able to do so, the client first has to learn about the service’s existence and its *.onion address. He can do so on specific web pages in the Dark or Surface Web, or from friends. After the client is in procession of the address, he can download the descriptor from the hash table. The client is now aware of the introduction points, thus he can create a TOR circuit to a so-called Rendezvous Point (RP).

In the next step, the client sends a message to the onion service via the RP that contains a One-Time-Secret (OTS), and the address of the RP, and is encrypted with the onion service’s public key. The RP forwards it to one of the introduction points. The OS responds to the client’s message by creating its own TOR circuit to the rendezvous point and by sending a message with the client’s OTS to this point. After the RP received and verified this message, he notifies the client that the connection is established [23] [61]. Fig. 2.6 illustrates this procedure.
2.3.2 Pretty Good Privacy

PGP is often used within the Dark Web, especially within Dark Web marketplaces and forums. It was developed by Philip Zimmermann in 1991 and is primarily used to encrypt and sign emails. However, in the Dark Web, it is also often used to encrypt or sign private messages, or files to secure communications and/or transactions in the Dark Web [19]. For encryption, the RSA algorithm can be used as well as the Diffie-Hellmann algorithm since Version 5.0. RSA is also used for signing messages and Message Authentication Codes (MACs).

In some Dark Web forums, users can publish their public PGP keys in their user profiles, in others it is common to post them in a specific thread. These published keys allow all forum users to start an encrypted communication with the key owners. As a PGP key (if not stolen) should always belong to a specific owner, they can also be used to identify users. Hence, PGP keys will be used as ground truth to link users within different forums in this thesis and thus are of significant importance.
Chapter 3

Artificial Intelligence

In this thesis, two sub-categories of Artificial Intelligence (AI) mainly are used, Natural Language Processing (NLP) and Machine Learning (ML). Both are essential for this thesis, and therefore, all needed theoretical foundations are explained in more detail in this chapter.

3.1 Natural Language Processing

By using NLP, computers can extract and understand information derived from language. It can be applied to text as well as speech and is an important part of AI [64]. It can be used for spell-checkers, within customer services, or in assistants like Siri or Cortana [32]. If no other reference is mentioned, the following explanations are based on [64].

3.1.1 Basics of NLP

To understand NLP, it is important to know how natural language is composed. This composition is not based on any artificial procedures that a machine has accomplished. It was rather developed, and naturally evolved by humans. Furthermore, the nature of meaning, as well as the usage and cognition of a language, is very im-
important. However, it is also essential to know how humans perceive, understand, and learn natural languages. With this knowledge, it is possible to transfer the learned concepts into algorithms that can be used within computer programs.

### 3.1.1.1 Linguistics

Linguistics focuses scientifically on syntax and semantics of a language based on the way it is used in, and in which context. It can be dated back to the 4th century BCE and is a huge field of scientific research. However, for this thesis, there is no need to understand all of it. Hence, only those subfields that are important for this thesis, will be described below.

- **Syntax**: the study of the structure of words, sentences, or phrases is called syntax and is extremely relevant as a change of the order of words, can change the entire meaning of a sentence.

- **Semantics**: the field of semantics can be divided into lexical semantics and compositional semantics. The former denotes the study of meanings of single words and symbols. In contrast, the latter denotes the study of relationships between words as well as within a combination of words. Additionally, the meaning of whole phrases and sentences is considered in this category.

- **Stylistics**: this subfield focuses on analyzing the writing style of a writer.

### 3.1.1.2 Syntax and Structure

Syntax and structure of a language are based on rules as well as conventions that denote in which way words, phrases, and clauses are combined to build a sentence. These rules and conventions differ between languages, but the main concepts (which are described in the following for the English language), also appear in other languages.
Fig. 3.1. Lots of words without order or relation from [64]. This example illustrates the difficulty of understanding the semantics of a sentence without correct syntax.

**Fig. 3.2.** In contrast to Fig. 3.1 this time the words are ordered and structured by following the hierarchical syntax as used in NLP to build an understandable sentence [64].

The sentence “*The brown fox is quick and he is jumping over the lazy dog.*” contains many words that hardly make sense when their order is incorrect as in Fig. 3.1. Words are the smallest unit in this sentence. Combining words leads to phrases, and by using conjunctions, clauses can be built that create the final sentence. When considering the correct order, as well as the syntax hierarchy (sentence, clause, phrase, word), as shown in Fig. 3.2, the sentence becomes understandable again.

As shown in Fig. 3.2 a sentence is made out of several elements. Words are the most obvious of these elements and can be categorized into several syntactic categories like nouns, verbs, adjectives, adverbs and many more. These categories in turn are called Parts of Speech (POS) and have significant importance within many NLP applications. Another basic element of a sentence is a phrase, a combination of words that belong to a meaningful phrasal category like noun phrase, verb phrase, adjective phrase, etc. A phrase consists of one main/headword and others that are grouped around it. Shallow parsing is a technique that can extract POS categories.
from words as well as phrase categories and thus support text analytics. In contrast to the previous two elements, clauses, another element of a sentence, can either be independent and act as a full sentence, or they can be dependent, which means that they need other clauses to create a meaningful sentence. However, clauses can be used for various purposes. For instance, some clauses are declarative, which means that they simply express a statement without any other intention. Another important aspect of a sentence is the grammar that was used to structure it. The grammar of a language defines the order of words, phrases, and clauses within a sentence. The rules that are used vary from language to language but also from region or dialect within a single language.

### 3.1.2 NLP Techniques

In this section, the focus is on different techniques that can be used to process text to prepare it for subsequent text analysis. These techniques can be divided into two main categories: on the one hand, in techniques that process the text in such a way that it is machine-understandable and contains as little noise as possible; and on the other hand, in those that can be used to extract features from the processed text, which are the basis for text analysis.

#### 3.1.2.1 Preprocessing

Texts do not always have the same structure, are not always formatted appropriately, and usually contain distortions that can influence the final result of text analysis. For this reason, a text must first be prepared for machine processing before it can be analyzed.

**Removing HTML Tags:** texts obtained using, e.g., a web crawler, do not only contain the desired text, but also maybe HTML, JavaScript or similar code snippets.
These will impurify the text and therefore have to be removed before starting the analysis. For this purpose, special HTML parser libraries can be used, which will remove most of the HTML code fragments. However, in special situations, the parser has to be customized to the given HTML files.

**Tokenization:** a tokenizer divides a given text into smaller pieces. These pieces can be either sentences or words, which are needed by other NLP techniques like stemming or tagging.

**Spelling errors:** in some applications, spelling mistakes are corrected in a preprocessing step. This is to ensure that the words can be understood, e.g., by a tagger, in a later step. Incorrectly written words, or words that may have been intentionally written with many double letters, can only be recognized and corrected by a self-written program. Because of that, and since this task can quickly become very complex depending on the accuracy with that it is performed, it should be carefully considered whether the expected benefit is high enough so that it outweighs the efforts.

**Stemming:** to be able to understand what stemming is, it is necessary to know what word stems are. The smallest lexical unit that exists is not a word but a morpheme. Morphemes consist of stems as well as affixes. The latter is an umbrella term for prefixes, suffixes, etc. that are prepended/appended to words. A word stem, on the other hand, can be understood as the base of a word. If affixes are attached to it, its meaning can change. This process is called inflection. However, stemming refers to the process of extracting the root of a given word instead of appending parts of a word to the root. An example would be the word jumping from which someone wants to extract the stem jump. In Fig. 3.3 a stem and some possible affixes are visualized. Stemming serves to normalize texts and helps clustering algorithms or
classification algorithms to analyze words independently of their affixes.

**Lemmatization:** lemmatization is a very similar process to stemming. However, it differs in one essential characteristic. In the case of stemming, the stem of a given word is being sought and no attention is paid to whether this stem is the root of the word, i.e., a grammatically correct word. As an example, the word stem of the word *studies* is “studi”, which is not a gramatically correct word, whereas its lemma “study” is grammatically correct. This is different from lemmatization and therefore this process needs much more computing time than a simple stemmer.

**Stopwords:** the term stopword refers to all words that occur very frequently in a language. Typical examples of stopwords in the English language are the words “a”, “the”, “and”, etc. These words are completely unsuitable for text analysis since they are used extremely frequently in all texts so that they do not contribute to the recognition of a significant difference between texts. Therefore, in most cases, they are removed before text analysis.

**Tagging:** during parts of speech tagging, each word is assigned with its abbreviation depending on its category. For example, if a word is a noun (singular), it gets the tag NN, if it is a verb, it gets the tag VB. There are 36 different tags that are based on the Penn Treebank Project that can be assigned to words within a text by using the nltk module for Python (see [1]). Another option is to use the spacy
module for python to extract tags as shown in [2].

### 3.1.2.2 Feature Engineering

After preprocessing, the text follows the next important step to a successful text analysis, the extraction of features. Machine learning techniques and especially deep learning models have a hard time processing text. Therefore, good feature engineering is essential to extract meaningful features from the given text and convert them into numeric values. The easiest way to convert more or less structured text into a machine-readable form is to use vector space models. These models transform text into numerical vectors using mathematical-algebraic methods. The dimensions of these vectors depend on the technique used to transform the text. These techniques are described in more detail below.

**Bag of Words Model:** the Bag of words model belongs to the traditional count-based feature engineering techniques. It uses term frequencies which means, it counts how often which words occur in which text. The resulting Vector Space (VS) model can be described mathematically as an $m \times n$ matrix, where $m$ describes the number of different text documents, and $n$ the number of distinct words within all texts of the text corpus. The name bag of words originates from the fact that the model does not consider syntax, word order, or semantics. It only focuses on the number of different words existing in a text corpus.

Besides the bag of words model, there is also a bag of N-grams model. The principle remains the same, only the words are replaced by character or word n-grams. For example, "How are you?" can be divided into the character-3-Grams 'How', 'ow_', 'w_n', '_ar', 'are', 're_', 'e_y', '_yo', 'you', 'ou?'.

The normal bag of words or n-gram models can cause problems if the text corpus is very large. Some words (or n-grams) may occur very often within a corpus, while others appear only in very few texts. This can lead to overshadowing the rarer
terms. But especially these words, which occur very rarely, are particularly suited as features to classify texts. To solve this problem a TF-IDF model [64] can be used instead of the pure Term Frequency (TF) model. The frequency by which a term \( w \) occurs in a document \( D \) is then multiplied by the Inverse Document Frequency (IDF). The \( IDF(w, D) \) is calculated as follows:

\[
IDF(w, D) = 1 + \log \frac{N}{1 + df(w)}, N = |documents|
\]

where \( df(w) \) refers to the document frequency of \( w \).

**Sentiment:** the analysis of the mood or sentiment of a text is a very popular field within NLP. It is also known as opinion analysis and it is based on various techniques from the area of text analytics, NLP, machine learning, and linguistics to extract information that can be used to make a statement about whether e.g., a text is written in a positive, negative or neutral way. Such a result is based on sentiment polarity which is a numeric value assigned to positive as well as negative aspects of a written document. This, in turn, is based on subjective parameters such as specific words and phrases, which are expressing feelings as well as emotions. There are several different techniques for sentiment analysis like unsupervised lexicon-based models, more traditional supervised machine learning, but also newer or advanced deep learning models [64]. In this thesis, only one representative from the first category is used.

**Word2Vec:** classic feature engineering methods like bag of words and TF-IDF have a major disadvantage. This is their simplicity, to be precise, it is the fact that they only consider the occurrence of words but not their structure, order, or semantics. This is the motivation for more complex feature engineering models that represent this missing information in a vector representation of words. This way of
representation is also called *word embeddings*. The basic idea is that a word is not characterized by its frequency within a text, but by its context. One of the most popular of these models is called the Word2Vec model and was developed by Tomas Mikolov in 2013 at Google [47]. It is a deep learning-based model and generates dense vector representations of words in which semantic similarities, syntactic similarities, and relation with other words are included.

There are two different versions of the Word2Vec model. This is the Continuous Bag of Words (CBOW) model and the skip-gram model which are both based on neural networks. When either of these two models should be used can only be evaluated by experiment. In most cases, the CBOW model is faster, but the skip-gram model is more accurate. The gensim framework for Python offers an implementation of word2vec that can work efficiently and robustly even on large datasets [64].

**GloVe and FastText:** besides the well known Word2Vec model, there are also some other models for creating word embeddings. These include, the Global Vectors for Word Representation (GloVe) model, developed in 2014 by Pennington et al. at Stanford University [58] and FastText, developed in 2016 by Facebook [18]. Like Word2Vec, GloVe is an unsupervised learning model but it is not based on neural networks but on a word-word co-occurrence matrix. The basic idea behind GloVe is to combine the two best-known methods to date, global matrix factorization (this category includes Latent Semantic Analysis (LSA)) and local context window methods (this category includes Word2Vec). FastText is an improvement of the Word2Vec model, which uses character n-grams in addition to each word. This idea is based on the fact that rare words of languages with a rich vocabulary are ignored in a conventional Word2Vec model. If all words are split into character n-grams, there is a higher chance that at least some of the character n-grams of rare words will appear within other words of the corpus and will not be ignored anymore because they appear too rarely [64].
3.1.2.3 Topic Modelling

In the following two paragraphs, two well-known topic modelling techniques are briefly explained. Both of them will be used to extract features, i.e., topics from each post within the dataset.

**Latent Dirichlet Allocation:** the Latent Dirichlet Allocation (LDA) \[17\] is one of the most popular topic modeling techniques. The algorithm is illustrated in Fig. 3.4 and is based on Eq. (3.2). \(W\) denotes words within a document, \(Z\) the per-word topic assignment, whereas \(\theta\) denotes the per-document topic proportions and \(\varphi\) the per-topic word distributions. \(\alpha\) is a dirichlet parameter and \(\beta\) a topic-parameter. All parameters are also illustrated in Fig. 3.4 except of \(z_{j,t}\) which is nothing other than the topic for the \(j\)-th word in document \(i\). The basic idea of the algorithm and Eq. (3.2) will be explained in the following paragraph without going in too much detail (for more information see Appendix [A.1] or the original paper [17]).

\[
P(W, Z, \theta, \varphi; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j)P(W_{j,t}|\varphi_{j,t}) \quad (3.2)
\]

The first step of the LDA algorithm is to initialize all necessary parameters (choose the number of topics, etc.). After that, it iterates over all documents and matches each word randomly to one of the \(K\) topics. The last step will be repeated several times to improve the results in each iteration. The algorithm computes \(P(T|D)\), the proportion of words within document \(D\) that is assigned to topic \(T\), \(P(W|T)\) which is how often the word \(W\) within all documents is assigned to topic \(T\), and finally, it reassigns each word \(W\) that is matched to topic \(T\), with the probability \(P(T|D) \times P(W|T)\) (which includes all other words and their respective topic assignments) [64].
Fig. 3.4. The input (left), output (right) as well as the basic principle of the Latent Dirichlet Allocation (middle) to extract topics from a text corpus based on Blei et al. [17].

**Non-Negative Matrix Factorization:** the second topic modeling technique used in this thesis is called Non-Negative Matrix Factorization (NMF). As the name indicates, NMF tries to find two non-negative matrices $W$ and $H$ so that their product equals a given non-negative matrix $V$ as best as possible [65]. The mathematical notation of the given problem is shown in Eq. (3.3).

$$V \approx WH$$ (3.3)

The best way to achieve the desired approximation is to use a cost function. An example of such a function is the Euclidean distance, the L2 norm or the Frobenius norm between two matrices. The mathematical representation can be seen in Eq. (3.4) [64].

$$\arg\min_{W,H} \frac{1}{2} \| V - WH \|^2 = \frac{1}{2} \sum_{i,j} (V_{ij} - WH_{ij})^2$$ (3.4)

In this thesis, the scikit-learn implementation of NMF (and Eq. (3.4) respectively) is used, which can be found in the scikit-learn decomposition module [65].
3.2 Machine Learning

Machine Learning (ML) is one of the most important current trend topics in the field of digitalization [73]. A growing number of companies are collecting immense amounts of data to analyze it as profitably as possible. ML is an instrument to identify patterns in data sets and to derive prognoses on this basis [51]. The main issue when using machine learning classifiers is that there is no perfect classifier that can deliver the best results under all circumstances. Each classifier/algorithm has its strengths in different areas and its weaknesses in others. It always requires systematic experiments to find the best classifier for a particular problem. This issue is also described by the “no free lunch” theorem. This fact has been proven by mathematicians [39]. A detailed description of the Theorem can be found in [78]. In the following, three classifiers that are used within this thesis are explained as well as some basics that are needed to understand and evaluate the choice of classifiers as well as their results.

3.2.1 Basics of ML

In the following subsections, a brief introduction to the basics of machine learning (variables, types of algorithms, evaluation criteria, etc.) is given.

3.2.1.1 Categories of Variables

ML-algorithms operate based on large amounts of data whose structure can be completely different. Thus, there are at least 4 types of variables to distinguish according to Ng and Soo [51]. Binary variables belong to the first category and have only two possible values. The next category contains all qualitative variables, which includes all variables with more than two options, but which cannot be represented as a number, such as animal species. Discrete variables belong to another category. They are always numerical and have a countable number of values between two numbers, such
as integers. The last category contains all continuous variables, that have an infinite number of possible values between two numbers. For example, 13.11 is between 13 and 13.5 but also 13.4111, or 13.333, and so on.

### 3.2.1.2 Selection of an Algorithm

When selecting a suitable algorithm for the available data, the type of data and the problem to be solved should be considered in advance. According to Ng and Soo [51], three categories help to find a suitable algorithm. The first is called unsupervised learning where data is fed into an algorithm/classifier without knowing which samples belong to which classes. It is the responsibility of the algorithm to find a pattern in this data. The opposite category is supervised learning where the class of a sample is known in advance. Thus the algorithm needs to find a pattern within the training samples of each class, that describes this class most appropriately. The last category, Reinforcement Learning, differs from the two previous categories in that the learned model is consistently improved. This is done by using the results as feedback.

**Choice of Parameters:** many parameters of a machine learning classifier can be optimized so that the same algorithm can produce different results, based on the same training data. The effects that can occur when choosing parameters are briefly explained in the following. A model is overfitted, when it can provide very accurate and detailed information about the training data, but is not suitable to classify future data that was not included in the training data set. The opposite effect is called under-fitting. A model is under-fitted when it is neither able to classify the training data correctly nor future data. The most desirable effect is called the ideal fit. If the parameters of a classifier have been set correctly, then the resulting model can detect the most important information, but cannot be distracted by smaller fluctuations or outliers [51].
3.2.1.3 Evaluation of the Results

The following explanations of the different evaluation criteria and basic quantities are based on Kubat [39].

- **The Basic Quantities:** there are four different quantities. An element is called *True Positive (TP)* when it is positive and also classified as such. In the opposite case, an element is called *True Negative (TN)* when it is negative and also classified as such. When an element is negative, but is incorrectly classified as positive, then it is called *False Positive (FP)*. The last quantity is called *False Negative (FN)* and denotes an element that is positive, but that was incorrectly classified as negative.

- **Error-Rate (E):** the error rate indicates how many errors an algorithm has made with respect to all given examples.

\[
E = \frac{N_{FP} + N_{FN}}{N_{FP} + N_{FN} + N_{TP} + N_{TN}}
\]  

(3.5)

- **Accuracy (Acc):** accuracy is the counterpart of the error rate \( Acc = 1 - E \). It indicates the frequency of correctly classified elements over all examples.

\[
Acc = \frac{N_{TP} + N_{TN}}{N_{FP} + N_{FN} + N_{TP} + N_{TN}}
\]  

(3.6)

- **Precision (Pr):** especially in the case of very unbalanced datasets (e.g. 2% positive and 98% negative elements), the error rate or accuracy, does not give much information about whether a classifier is appropriate or not.

The precision averages the proportion of elements identified as true positive over all elements that were identified as positive. Thus it states how often a
classifier is correct when it identifies an element as positive.

\[ Pr = \frac{N_{TP}}{N_{FP} + N_{TP}} \]  

(3.7)

- **Recall (Re):** it indicates the probability that a positive element will be identified as positive by the classifier.

\[ Re = \frac{N_{TP}}{N_{TP} + N_{FN}} \]  

(3.8)

Whether precision or recall is more important for classification always depends on the application. If, for example, a company wants to present personalized advertising to a customer, it is probably more interested in a much higher precision than in a high recall. This is because a customer ignores advertising if he does not like it. A high recall rate might be important in medical applications. A patient that is identified as healthy but is ill (false negative), is something that a doctor wants to avoid.

- **F\_β score:** the F\_β score combines the values for precision and recall in one value.

\[ F_\beta = \frac{(\beta^2 + 1) \cdot Pr \cdot Re}{\beta^2 \cdot Pr + Re}; \quad F_1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \]  

(3.9)

The parameter \( \beta \in [0, \infty) \) indicates how much weight is given to the recall and how much is given to the precision. \( \beta > 1 \) denotes more weight towards recall, whereas \( \beta < 1 \) denotes more weight towards precision and \( \beta = 1 \) indicates a balanced ratio. Since in many cases it is not clear which of the two values is more important, the \( F_1 \) score is used most frequently.

**N-Fold Cross-validation:** in advanced statistical evaluations, the principle of n-fold cross-validation is frequently used. The idea of this principle is to split the entire data set up into n equal parts (sets). Thereby n-1 parts are always used for
training and the remaining part for testing. This procedure is performed $n$ times, always with changing sets for training and testing. In the end, the average and the standard deviation are calculated from all results (example in Fig. 3.5).

### 3.2.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning model. It was developed by Wladimir Wapnik and Alexei Jakowlewitsch Tscherwonenki [76], [77] based on the simple linear model developed in 1957, by Frank Rosenblatt [63]. In 2D, a linear model is a line (in higher dimensions a hyperplane) that separates data from two different classes. Starting from the assumption that the given data can be separated by a line, it is known that this line can be described by $y = wx + b$. Furthermore, it is known that there is an infinite number of possibilities for the choice of the parameters $w$ and $b$. Therefore, an algorithm like Support Vector Machine is needed that determines the best values for the two parameters [38].

#### 3.2.2.1 Fundamental Functionality of two-class Support Vector Machines

During the training of a two-class Support Vector Machine, a decision function needs to be found that separates the two given classes so that their distance is maximized. In this way, Support Vector Machine achieves a very high degree of generalizability.
3.2.2 Hard-Margin Support Vector Machines

The following explanations are based on Abe \[11\] if no other source is mentioned. Be \(x_i, (i = 1, \ldots, n), x \in \mathbb{R}^m\) \(n\) m-dimensional training data samples belonging to either class one or two. Be \(y\) the label for these two classes with \(y_i, (i = 1, \ldots, n), y \in \{-1, 1\}\) so that \(y_i = 1\) if \(x_i \in\) class one and \(y_i = -1\) if \(x_i \in\) class two.

In the case of hard-margin Support Vector Machines, the training data can be easily separated by a line or in general by a hyperplane. However, there are many different ways to arrange this hyperplane between the two classes. The trick of Support Vector Machine is to find the hyperplane that separates the two classes and has the largest

![Hard Margin SVM and Soft Margin SVM](image)

Fig. 3.6. Graphical representation of the difference between hard-margin (left) and soft-margin (right) Support Vector Machines according to MLMath.io [48]. In contrast to hard-margin Support Vector Machines, soft-margin Support Vector Machines are able to correctly classify at least some of the depicted outliers (see Appendix A.2). and reduces the chance of overfitting. In the following, the mathematical background of hard-margin Support Vector Machines is described. With these Support Vector Machines, the training data can be linearly separated. The counterpart to hard-margin Support Vector Machines are soft-margin Support Vector Machines for linearly inseparable data (for more information see Appendix A.2 or [11]). A graphical illustration of the difference between these two types of Support Vector Machines is given in Fig. 3.6.
distance to both of them. But how can this position be calculated? To solve this problem, a solution of a decision-function \( D(x) \) must be found.

\[
D(x) = \mathbf{w} \cdot \mathbf{x} + b,
\]  

(3.10)

where \( \mathbf{w} \) is a normal vector and \( \mathbf{w} \cdot \mathbf{x} \) is nothing other than the scalar product of \( \mathbf{w} \) and \( \mathbf{x} \). In this case, \( \mathbf{w} \) is an \( m \)-dimensional vector and \( b \in \mathbb{R} \) is a scalar (see Eq. (3.17)). For all \( i = 1, \ldots, n \) the following inequalities are considered:

\[
\mathbf{w} \cdot \mathbf{x}_i + b \begin{cases} > 0 & \text{if } y_i = 1, \\ < 0 & \text{if } y_i = -1. \end{cases}
\]  

(3.11)

\( \mathbf{w} \cdot \mathbf{x} + b = 0 \) describes the hyperplane, and \( \mathbf{w} \cdot \mathbf{x} + b = 1 \), as well as \( \mathbf{w} \cdot \mathbf{x} + b = -1 \) describe the borderlines of the maximum distance on the left and right side of the hyperplane between the two classes one and two (see Fig. 3.7). In this section, the focus is on linearly separable data, which means that the data points of the different classes reach a maximum distance of \( a \) to the hyperplane. This distance is calculated using support vectors, which are all data points that are closest to the hyperplane. In Fig. 3.7 there are two support vectors from the blue class and two from the red class. Thus, these support vectors can be used to calculate two boundary lines with the hyperplane located in the middle.

Since at this point it is assumed that the training data can be separated linearly, according to Abe [11] the conditions from Eq. (3.11) can also be slightly transformed to:

\[
y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1
\]  

(3.12)

Thus, the two conditions of Eq. (3.11) for all \( i = 1, \ldots, n \) can be reduced to one condition [36]. However, with this inequality alone, it is still not possible to answer
Fig. 3.7. The hyperplane with a maximum distance class a (blue) and class b (red).

the question of how to find the hyperplane with the greatest distance between the
two classes. A general formula for the hyperplane is known (see Eq. (3.10)) but not
which vectors $x_i$ for $i = 1, ..., n$ of the training set are support vectors. Therefore,
a mathematical description for the distance or margin $m$ of the hyperplane to the
support vectors is required and can be found in Eq. (3.13) [36].

$$m = \frac{2}{\|w\|}$$  (3.13)

However, this margin still has to be maximized. When inserting a few example
values for $\|w\|$ into Eq. (3.13), it can be seen that the larger these inserted values
become, the more the value of the distance $m$ decreases. From this experiment, it
can be concluded that maximizing the distance has the same effect as minimizing
the norm of $w$. Since it is the goal to maximize $m$, the hyperplane with the smallest
value for $\|w\|$ is wanted [36]. With this in mind the following optimization problem
can be set up [37]:

$$\text{minimize } f(w, b) = \|w\| = \frac{1}{2}\|w\|^2$$  (3.14)

subject to $y_i(w \cdot x_i + b) \geq 1$ for any $i = 1, ..., n$  (3.15)
Note that there are $n$ different conditions in Eq. (3.15) and not just a single one as might be assumed at first sight. To not go into too much detail, $w$ and $b$ can be found after solving an optimization problem that can be either transferred to a dual (maximization) or primary (minimization) problem. In the end, the following equations for $w$ and $b$ can be obtained.

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (3.16)$$

$$b = \frac{1}{S} \sum_{i=1}^{S} (y_i - w \cdot x_i), S = |\text{support vectors}| \quad (3.17)$$

where $\alpha_i$ is the also called the Lagrange multiplier that originates from solving the optimization problem with constraints like Eq. (3.15). In the end, a training data point $x$ is assigned to class one if $D(x) > 0$ and to class two if $D(x) < 0$. If $D(x) = 0$, then the data point $x$ is on the hyperplane and cannot be classified.

$$D(x) = \sum_{i=1}^{S} \alpha_i y_i x_i \cdot x + \frac{1}{S} \sum_{i=1}^{S} (y_i - w \cdot x_i) \quad (3.18)$$

3.2.2.3 Fundamental Functionality of multi-class Support Vector Machines

The operating principle of Support Vector Machine, as described in Section 3.2.2, is designed for a two-class problem only. However, to overcome this issue, there are four main approaches to apply Support Vector Machine on data with more than two classes. The first approach is called “one-against-all” Support Vector Machines. Its idea is, to transform an $n$-class problem into $n$ two-class problems and then apply the normal Support Vector Machine operating principle on these $n$ problems. Unfortunately, this approach has one drawback. By using the discrete decision functions of the original Support Vector Machine algorithm, unclassifiable areas might occur. Another approach, called “pairwise” Support Vector Machines, tries to solve this
difficulty by transforming an $n$-class problem into $\frac{n(n-1)}{2}$ problems instead of just $n$ problems as before. Thus, this approach covers all possible pairs of classes, but unfortunately, even in this case, unclassifiable areas might exist. However, a solution for these unclassifiable areas might be the usage of “membership-functions”, decision trees, or error-correcting output codes. The latter one is applied by the third approach, the Error-Correcting Output Code (ECOC) Support Vector Machines. The last type of multi-class Support Vector Machines are the “all-at-once” Support Vector Machines. In this approach, all decision functions are computed simultaneously.

### 3.2.3 Decision Trees

The following explanations are based on Kubat [39] as long as no other source is mentioned. The procedure of classifying data is completely different when using DTs than when using an Support Vector Machine classifier. Support Vector Machine tries to find a hyperplane that splits all training data at once, whereas DTs use a stepwise classification approach. A DT consists of several nodes, leaves, and edges. Each node of the tree represents a feature, the edges represent possible characteristics of the feature, and leaves represent the classes. If a new sample has to be classified, the feature at the root of the tree is checked first. From that point on, the algorithm follows the edges that correspond to the values of the features present in the sample until a leaf is reached. This leaf specifies to which class the sample belongs.

A big advantage of decision trees over e.g., neural networks or Support Vector Machines is their explainability. Sequentially going through the tree starting from the root going to the leaf, explains how the algorithm came to its result.

The simplest explanation of how a DT can be created from training data is the principle of divide and conquer. In this way, the features of the input data are split up starting from the root until all instances can be correctly assigned to their class.
However, this process can result in a large number of different trees, which all deliver correct results. The question is, which of these trees is the best and how can this be determined? The size (number of nodes) of the tree is certainly the simplest criterion. But also the average number of nodes (equal to the average number of tests a sample has to pass) plays a major role. However, there are several reasons why small trees are preferred. On the one hand, the explainability becomes more difficult the bigger the tree becomes. On the other hand, smaller trees usually contain less redundant or irrelevant information. Except that, large trees tend to be over-fitted and thus unable to classify unknown test data properly.

3.2.3.1 Information Content of Features

The following explanations are based on a simplified version of the C4.5 algorithm. There are other algorithms to build a DT besides this one, but which will not be discussed here.

Except the size of the tree and the average number of tests, the amount of information a feature contains also plays a major role in the development of a DT. For example, there may be features that can be used to immediately determine which class the instance belongs to. However, there might be also features that only lead to a unique classification if other features are taken into account. Therefore, an algorithm to measure this “information content” of the individual features is needed. There are two basic approaches, the first is called “Gini impurity” and the other “Information Gain Entropy”. The formula for measuring the Gini impurity is:

\[
Gini(p) = 1 - \sum_{j=1}^{c} p_j^2
\]  

(3.19)

where \( c \) is the number of classes within the dataset and \( p_j \) denotes the fraction of samples belonging to class \( j \) \([62]\). However, the approach “Information Gain
Entropy” will be explained in more detail in the following paragraphs.

To start the calculation of a DT, it must be known which sample belongs to which class. From this, the relative frequencies of the positive (pos) class \( (p_{pos}) \) and the negative (neg) class \( (p_{neg}) \) can be determined. \( p_{pos} = 1 \), if all samples belong to class pos, \( p_{pos} = 0.5 \), if both classes occur equally frequently and \( p_{pos} = 0.03 \), if the large majority of samples belong to the negative class. To determine how much information \( I \) a feature contains, the following formula can be used:

\[
I_{pos/neg} = -\log_2 p_{pos/neg}
\]  

(3.20)

To determine the entropy of a training set \( T \), Eq. (3.21) needs to be calculated. However, to calculate the information content of a feature, Eq. (3.21) needs to be used with respect to the corresponding values of a feature within a class (see Eq. (3.22)).

\[
H(T) = -p_{pos}\log_2 p_{pos} - p_{neg}\log_2 p_{neg}
\]  

(3.21)

\[
H(T_i) = -p_{pos}\log_2 p_{pos} - p_{neg}\log_2 p_{neg}
\]  

(3.22)

Eq. (3.23) indicates the probability that a randomly selected training sample contains the attribute \( T_i \).

\[
P_i = \frac{|T_i|}{|T|}
\]  

(3.23)

Using Eq. (3.22) and Eq. (3.23), a formula can now be created that specifies the weighted average of the entropy values of all possible values of a feature \( fe \).

\[
H(T, fe) = \sum_{i=1}^{|T|} P_i H(T_i)
\]  

(3.24)

To calculate the information content of feature \( fe \), Eq. (3.25) can be used. In this formular the entropy of feature \( fe \) is subtracted from the entropy of the training set.
without considering feature $fe$.

$$I(T, fe) = H(T) - H(T, fe)$$  \hspace{1cm} (3.25)

A brief description of how continuous values are transformed so that they can be used in a decision tree can be found in Appendix A.3.

### 3.2.4 Naive Bayes

Already before World War 2, scientists attempted to predict the class of a sample by using the Bayesian probabilistic theory. For each class, they have calculated the likelihood that the sample belongs to it, whereas in the end, the class with the maximum likelihood is picked. The Naive Bayes classifier is based on the same principle and will be explained in the following sections in more detail. All explanations are based on Kubat [39].

#### 3.2.4.1 Probabilities

Different types of probabilities play an important role in the idea of Naive Bayes. The first is the prior probability $P(pos)$ or $P(neg)$ that refers to the likelihood that an attribute is either positive or negative respectively. It can be computed by:

$$P(pos) = \frac{N_{pos}}{N_{all}}, \text{ and } P(neg) = \frac{N_{neg}}{N_{all}},$$  \hspace{1cm} (3.26)

where $N$ denotes the number of samples. In this case, only attributes that can be either positive or negative are considered. If a second attribute (e.g. a color) is added to the data set, the joint and conditional probabilities can be calculated. The former can be written as $P(pos, red)$ and denotes the likelihood, that a sample is positive and red. This should not be mixed with the latter one. The conditional probability $P(pos|red)$ indicates how likely it is to find a positive example among
all those that are red. Interestingly, the joint probability can be obtained from the prior probability and conditional probability.

\[ P(pos, red) = P(pos|red) \cdot P(red) \]  

(3.27)

Hence, by transforming this equation, the conditional probability can also be derived from the joint probability. As \( P(red, pos) \) and \( P(pos, red) \) describe the same thing, they have to be equal. Thus:

\[ P(pos|red) = \frac{P(red|pos) \cdot P(pos)}{P(red)} \]  

(3.28)

### 3.2.4.2 Discrete Attributes

Datasets that contain only two attributes are not very common in the real world. Hence, the Bayes formula needs to be adjusted in a way, so that it is capable to deal with multiple attributes, i.e., vectors like \( x = (x_1, ..., x_n) \), that describes all \( n \) attributes of a given sample. Furthermore, multi-class problems are more realistic than a simple two-class problem. Hence, given these two modification requests, the Bayes formula for a data set with multiple classes \( y_j, j \in \{1, ..., k\} \), and a vector \( x \) that describes the sample that has to be classified, has the following form:

\[ P(y_j|x) = \frac{P(x|y_j)P(y_j)}{P(x)}, j \in \{1, ..., k\} \]  

(3.29)

Therefore, for each class, the likelihood that samples described by vector \( x \) belong to class \( y_j \), is computed. Thus the denominator stays always the same, only the numerator is changing and needs to be maximized to find the most likely class for \( x \). The prior probability \( P(y_j) \) is relatively simple to estimate, by computing the relative frequency of \( y_j \) within the given training set. Unfortunately, it is not that easy to obtain the solution of \( P(x|y_j) \). The vector \( x \) might occur only once within
the whole training set, and thus $P(x|y_j) = 1$ in only one case and would be equal to zero in all other $k - 1$ cases. The situation would become even worse when a vector within the testing set does not occur within the training set. In that case $P(x|y_j)$ would be zero for all classes. Hence, the numerator would be zero for all classes and it would be impossible to classify the sample.

Therefore, the probability that attribute $x_i, \forall i \in \{1, ..., n\}$, where $n$ denotes the total number of attributes of vector $x$, will have a certain value, needs to be combined with the probability that $x$ occurs in class $y_j$. If all attributes are independent, then $P(x_i|y_j)$ denotes the likelihood that the value of attribute $i$ of class $y_j$ is $x_i$. Therefore, the probability that a random sample $x$ belongs to class $y_j$ is denoted as:

$$P(x|y_j) = \prod_{i=1}^{n} P(x_i|y_j), j \in \{1, ..., k\}$$ (3.30)

With that in mind, a sample will be classified as belonging to class $\hat{y}$ by computing the maximum likelihood that $x$ belongs to class $y_j$ by:

$$\hat{y} = \arg\max_{j \in \{1, ..., k\}} P(y_j) \cdot \prod_{i=1}^{n} P(x_i|y_j).$$ (3.31)

However, it is unlikely that all attributes in a dataset are independent. Therefore it is naive to assume this, and that is the reason why the Naive Bayes classifier is called naive. The violation of the constraint of independence might lead to an inaccuracy while estimating the probabilities. Fortunately, this does not have to lead to wrong class choices by the classifier, since the class is chosen by the maximum likelihood. When the probability of $x$ belonging to class $A$, or the likelihood that it belongs to class $B$ differs significantly, then the result will not change, even when the accuracy of the probabilities might not be 100% perfect. Thus, even when the attributes are not independent, the results will still be quite good. How Naive Bayes deals with continuous values can be found in Appendix A.4.
Chapter 4

Dataset

4.1 Creating a new Dataset

One of the most important parts of this thesis was to create a suitable dataset for an authorship attribution analysis. Thus posts from different Dark Web forums had to be extracted. This was done by using a self-developed web crawler/scraper written in Python 3.6. However, before the web crawling/scraping process could be started, existing Dark Web forums needed to be found first. Dark Web websites are not indexed by Google. However, there are some search engines within the Dark Web e.g., Torch but their results are often not usable. Instead of that there are several websites within the Surface Web or Dark Web that serve as link list for Dark Web websites (e.g., TheDarkWebLinks within the Surface Web or DarkFail within the Dark Web). Unfortunately, not all of these websites are trustworthy as they can either spread malware themselves or list phishing links instead of the correct links for different onion services. Due to the experience gained during this thesis, DarkFail seems to be an overall trusted website by the Dark Web Community. This website only lists links that were cryptographically signed by the owner of the onion service and directly sent to the owner of DarkFail.
The following sections explain how the Dark Web crawler/scraper used in this thesis was developed, what special requirements it had to fulfill to be able to crawl Dark Web websites instead of Surface Web websites, and after all, which forums were used for developing the dataset. In a final subsection, the HTML parser that was also developed within this thesis to be able to extract the posts from all forums, and to clean the data from HTML tags, is described.

4.1.1 Developing a Dark Web Crawler and Scraper

A crawler can be explained as a tool that opens URLs and searches for new, not already crawled URLs on each new website that it is coming through. If new links are found, they are stored and are crawled one after another at a later point in time. A web scraper copies the content of a website and saves the resulting data. In this thesis, crawler and scraper are strongly connected as the crawler extracts all URLs that point to a website that needs to be stored. For this part of this thesis Python version 3.6 was used on a virtual machine from VirtualBox version 6.0, with Ubuntu 18.04.3 LTS, 8 GiB RAM, 64-bit, and an Intel Core i7-2600K CPU with 3.40GHz.

4.1.1.1 Main Crawler Functionality

When developing a crawler, it is helpful to stay in an easy and well-known environment like the surface web, until all functionalities are successfully tested. The following paragraphs give a brief introduction of the most relevant functionalities of a normal web crawler that were also implemented for the Dark Web crawler.

**URLs:** the elementary data of a crawler are URLs. They allow the crawler to move forward and crawl a website page by page. However, the question is, how to obtain all links of a website. This is quite easy as websites are written in HTML and links are tagged with a “<a>” tag that contains a `href` attribute. There are specific
packages for Python like *HTMLParser*, or *urllib*, that can detect all these tags and their attributes very easily and return them as a set of links.

**Send a request to a website:** in the case of a Surface Web crawler, this is an easy task by using the package *requests* version 2.22.0 for Python. If needed, a “header” like a user-agent (a string that specifies e.g., the browser and the OS of a user), as well as cookies can be sent together in a request to a web server. The HTML response can be used for further analysis like extracting all URLs or it can be stored on the file system (which is the scraping functionality).

**Queues and backups:** queues and backups allow the crawler to remember what it already has done. More specifically, the queue contains all URLs that have already been found by the crawler, but that still needs to be crawled. On the other side, all URLs that already have been crawled, are stored in a set. Both data structures are extremely fast and thus suitable for fast crawling. The crawler checks whether a URL (that it has found on the website it is currently crawling), is already in the queue or the set. When a URL is not in one of these two data structures, it will be added to the queue. When the crawler has successfully crawled a URL, it will store the URL in the set and will delete it from the queue. However, the queue, as well as the already crawled URLs, should be stored periodically on the file system. Thus, in the case the crawler crashes, it would not be necessary to start the whole crawl from the beginning.

**Stay on the target website:** this is a very important goal to keep in mind when developing a crawler. Except for companies like Google, Bing, or Yahoo, it is quite uncommon to crawl the whole web. Most of the time, only specific websites are the target of a crawl. Thus, when crawling websites, it is very important to check all URLs that have been found whether they point to a subpage of the target website
or whether they point to a completely different website. In the latter case, the URL should never be stored in the queue.

**Multithreading:** multithreading is unavoidable when developing a fast crawler. This is because a single thread needs longer to crawl a whole website than multiple threads, and thus it is recommended to use multithreading. However, it should be noted that all activities of all threads have to be well coordinated. This means two threads should never crawl the same URL and the editing process of the backup files should be locked so that only one thread at a time can change those files.

### 4.1.1.2 Special Requirements for the Dark Web

To develop a crawler for the Dark Web is a little bit more complicated than for the Surface Web. This is because the way of connecting to a Dark Web website is completely different. On top of that, Dark Web websites tend to protect themselves from being crawled or attacked e.g., by a Distributed Denial of Service (DDoS) attack more rigorously than websites on the Surface Web.

**Establish a connection over TOR:** the TOR network is accessible over the TOR browser or a TOR proxy. For a crawler, the latter option is the most suitable solution. Thus, the Python package `stem` version 1.7.1 was used to establish a TOR circuit by using a proxy. After that, the normal `requests` package version 2.22.0 was used to establish a session over the given TOR circuit. In this thesis, multiple TOR proxies were set up to allow multiple TOR sessions in parallel. Thus each of these sessions had its TOR circuit that was renewed every ten minutes by using the `stem` package. This was done to simulate the typical behavior of the TOR browser (and thus to pretend to be a normal human TOR user and not a machine). In addition, the user-agent was set up with the real user-agent “Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0” extracted from a TOR browser to obfuscate
the crawler’s existence. Each version of the TOR browser uses just one single user-agent for all platforms. No matter if it is installed on macOS, Linux or Windows, the user-agent stays always the same except the version number. This was set up by TOR to provide more anonymity to its users as all their browsers look the same by keeping the user-agent the same. Therefore, choosing a completely different user-agent for each of the sessions (which would be typical for a Surface Web crawler) would make the crawling process detectable.

**Timing of requests:** this is an extremely challenging task when crawling onion services like Dark Web forums. Too many requests per minute might lead to a detection of the crawler, whereas too few requests will result in a slow crawl. Thus, a trade-off between both factors needs to be found. Unfortunately, not all Dark Web forums have the same protection mechanism against automatic requests like those from a crawler. As an example, the three forums The Hub (TH), The Majestic Garden (TMG), and DNM Avengers (DNMA) do not stop crawls with ten threads that send a request all 8 to 12 seconds each. On the other side, the very popular Dark Web forum Dread has an extremely strict DDoS protection that blocks the connection of a client that sends requests with a delay of less than approximately 5 seconds. On top of that, Dread also seems to keep an eye on the time interval of each connection. This means, when the requests of a specific client (in this case a thread of the crawler) sends multiple requests to the forum with the same time gap (e.g., a thread sends a request all six seconds), then the website can detect the thread as a bot. Thus, it is important to choose the right “minimum” amount of time, as well as an acceptable range of time to wait between the requests. As stated before, for TH, DNMA, and TMG an acceptable range was 8 to 12 seconds whereas for Dread 10 – 25 seconds worked best. The number of seconds a thread has to wait before starting a new request was chosen at random within these intervals. Unfortunately, it is a trial and error process to find the best time intervals for each forum, as it
would be self-damaging for Dark Web forums to publish any information about their bot detection settings. Therefore the ranges mentioned at this point can be seen as a guideline but do not have to work for all onion services.

**Adjusting forum settings:** to adjust the forum settings for each user account is very helpful. Most Dark Web forums do not allow users to access the content of the forum without being registered. Therefore, in three out of four cases, a user account was mandatory to be able to crawl the forum. Having an account on a Dark Web forum is often linked with the possibility to adjust the settings for the forum’s outward appearance. Thus, to reduce the number of pages that need to be crawled, the highest number of posts or threads shown per page was chosen. As multiple TOR proxies were set up in this thesis, also multiple user accounts were needed to be able to use the given benefit. Hence, one thread was set up with its own TOR proxy, its own circuit, its own user account but all of them were configured to use the same user-agent as a header.

**Blacklisting and whitelisting:** an approach that contains blacklisting as well as whitelisting was set up to avoid being logged out or accidentally send private messages during the crawling process. As an example, in the Hub, all URLs starting with `http://thehub***.onion/index.php?topic=, .../index.php?board=, or .../index.php?action=profile` were whitelisted. This means only URLs starting with one of these three prefixes were allowed to be crawled (and thus were stored in the queue or set). The blacklist contains words like `next, sort`, etc. which would have resulted in new URLs but the same content. These words are coming from the sorting functionalities which sort the order of posts or threads on a subpage. `PM or logout` are also typical examples of words within the blacklist that should never occur within a URL that is stored in the queue.
Table 4.1: Statistics of all four crawled Dark Web forums, showing an overview of
the dataset used in this thesis. The number of PGP-Key owners within TMG refers
to only those that could be matched.

<table>
<thead>
<tr>
<th>Forum name</th>
<th>Number of posts</th>
<th>Number of users</th>
<th>Number of users with PGP Key</th>
<th>Total number of files crawled</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNMA</td>
<td>75,165</td>
<td>10,489</td>
<td>277</td>
<td>20,645</td>
</tr>
<tr>
<td>TH</td>
<td>225,135</td>
<td>26,502</td>
<td>1,900</td>
<td>55,106</td>
</tr>
<tr>
<td>TMG</td>
<td>201,538</td>
<td>5,121</td>
<td>193+</td>
<td>15,678</td>
</tr>
<tr>
<td>Dread</td>
<td>385,839</td>
<td>63,299</td>
<td>2,943</td>
<td>163,943</td>
</tr>
</tbody>
</table>

Separating automatic and manual processes: unfortunately, this part was
indispensable as all forums require to solve a captcha when signing in, or logging in.
An example of such a captcha can be seen in Fig. 3.1. As these captchas are difficult
to solve automatically, the signup and login process was done manually. After the
accounts were set up properly, all cookies were extracted and added to the crawler.

4.1.2 Used Dark Web Forums

At the time of this analysis, there are less than 10 Dark Web forums that can be found
in popular Dark Web link lists like DarkFail, with a large and active community
(around 1000 authors or more). Unfortunately, the number of those forums that
allow users to publish their PGP keys in their user profiles (or those where users
post their keys in threads), is even smaller. Forums that are described in link lists as
specialized on e.g., child pornography, or assassinations, were neither accessed nor
crawled within this thesis due to the sensitivity of their content. Hence, only four
forums fulfilled the requirements for this analysis as they had a suitable number of
posts and users, as well as PGP Keys that could be linked to its owners.

DNM Avengers: this is the smallest Dark Web Forum that was crawled within
this thesis. It focuses mainly on drugs, but there are also some threads about more
Fig. 4.1. Screenshot of the start page of DNMA a smaller Dark Web forum [4]. Usernames, as well as time and dates, are blackened for privacy/security reasons.

general topics, politics, security, cryptocurrency or Dark Web marketplaces (a screenshot of the start page can be found in Fig. 4.1). For the registration in DNMA, a fake e-mail, a username, a password and solving a captcha was mandatory. As this forum was quite small, it took only around five and a half hours (wall time) to crawl it with 10 threads.

The Hub: TH is the “sister-forum” of The Majestic Garden. It is larger than DNMA with approx. 225,100 posts and 117,000 users in total. Unfortunately, only 26,500 users have posted at least one post (see Table 4.1). TH does not provide
a member list to normal users, thus only those 26,000 users that wrote a post are included in the dataset. As its name already indicates, TH is kind of a central point with lots of threads with discussions about other Dark Web websites, mainly marketplaces. Additionally, it contains lots of threads regarding security and vendor reviews. For the registration in TH a fake e-mail, a username, a password and solving a captcha was mandatory. However, it took around 34 hours to crawl this forum. In Fig. 4.2 a Screenshot of the first part of the start page is shown.
Fig. 4.3. Screenshot of the start page of TMG a Dark Web forum and marketplace [9]. Usernames, as well as time and dates, are blackened for privacy/security reasons.

**The Majestic Garden:** this onion service is a mixture of forum and marketplace. It has some threads dealing with general topics but the main focus is on drugs. It has boards for different kinds of drugs as well as drug safety boards, information about payment and shipping methods, as well as vendor recommendations, or vendor warnings. TMG has over 40,000 users in total but only around 5,000 that are active in the forum (wrote at least one crawled post). The registration process for a user account within this forum is a little bit special compared to the previous two. In addition to the same requirements mentioned for TH and DNMA, a PGP Key that has to meet various requirements needs to be sent within the registration form.
Fig. 4.4. Screenshot of the start page of Dread, one of the most popular Dark Web platforms [5]. Usernames, as well as time and dates, are blackened for privacy/security reasons.

Additionally, the registration has to be confirmed by one of the administrators which can take up to two days. On top of that, user profiles, in general, are not accessible by normal users which is a major drawback of this forum compared to the others.

Usually, PGP keys that are needed in this thesis to link the users between the forums are stored in these user profiles. However, as those are not accessible for any user except the administrators or moderators, users post their keys in specific threads so that they are public to the forum community. Therefore the linking process required a lot of manual work like checking whether a posted key seems to be posted by its
owner or whether someone else posted it on his behalf. Hence, the manual control process was reduced to only those keys that could be matched to a key within another forum. As user profiles were not crawlable in TMG, it only took around 26 hours to crawl this forum. In Fig. 4.3 a screenshot of the first part of the start page of TMG can be seen.

**Dread:** Dread is the biggest forum crawled within this thesis (see Table 4.1). In contrast to the others, it does not have a specific topic. It is more like a platform for everyone that wants to ask questions or talk about various topics. It recently faced lots of DDoS attacks and thus has an extremely strict DDoS protection which makes it difficult to crawl. Thus, the crawl was a huge cat and mouse game as the website owner continuously changed and improved Dread’s DDoS protection. Hence, the crawl was interrupted around a dozen times and took in total around 12 days (wall time). However, compared to the other forums, is contains private subthreads that are not accessible without a specific registration for this subthread. Thus, these specific threads could not be crawled. Fortunately, the whole forum can be crawled without creating a user account which is also a major difference to the other forums. An impression of Dread can be found in Fig. 4.4.

### 4.1.3 Developing an HTML Parser

The raw HTML data that was extracted by the web scraper is not suitable for text analysis. As described in Section 3.1.2.1 the text has to be cleaned from HTML tags and other code snippets. The following subparagraphs explain how the data needed for this thesis is extracted from the HTML files, and of course, which data is extracted.

**Extracting posts:** the first step was to separate all posts within a file from each other. In the case of TH, TMG and DNMA there can be up to 50 posts per page/-
file. Thus they need to be divided under consideration of the correct HTML “post-seperator” sequence. Therefore, a normal pre-implemented HTML parser that simply deletes all HTML tags could not be used as these tags were important to be able to separate the posts. It should be mentioned at this point that the initial/first post of a thread does not always have the same HTML structure as its replies. Thus, those posts have to be treated a little bit differently in most of the following steps.

**Extracting authors:** the username of an author is usually mentioned above or beside a post and thus is surrounded by undesired HTML code. However, simply removing this HTML code would make it a hard task to distinguish between the written text and the username as both of them would simply be concatenated. Therefore, HTML tags also help in this case to identify the position of the username within the post. Note that as mentioned before, these tags can be different between replies and initial posts and thus need to be parsed with caution.

**Extracting the time and date when a post was posted:** extracting this element of a post was especially complicated as DNMA and Dread had both a completely different time format than TH and TMG (which fortunately share the same format). However, this leads to three different time formats that needed to be transformed into a unique one. On top of that, Dread used the abbreviations “nd, rd, st, th” after the day of the week, which resulted in four possible ways in that time could be displayed. Additionally, initial posts did not contain a dash between the date that was mentioned first, and the time that was mentioned last. Thus, in the end, there were 8 distinct time formats only for Dread that needed to be parsed. Unfortunately, in DNMA as well as TH and TMG the date of a post that was written one day before the crawl, was displayed as “yesterday”. Additionally, the date of a post that was posted on the same day on that it was crawled, was displayed as “today”. Thus there were three time formats (the normal one, the one with yesterday, and
the one with today) for TH and TMG and three others for DNMA.

**Clean the posts from Quotes:** quotes contain content from an author that might not be the author of the post in that the quote was integrated into. Thus, they need to be deleted from an author’s writing as they might contain the text and writing style of someone else. When developing a parser for quotes it is important to keep in mind that: quotes can be included stand-alone within the post, interlaced, or after each other with the text of the actual author of the post in between. Hence, parsing quotes is extremely difficult and unfeasible without the consideration of HTML tags. Note, that quotes are non existent in Dread. Its outward appearance, as well as its structure, differs significantly from TH, TMG, and DNMA. It is more modern and thus users can reply directly to another reply of a post and do not need to quote it.

**Clean the posts of PGP keys or signatures:** the signatures and keys contaminate the text of an author with hex digits that do not have anything to do with the writing style or content of the original text. Thus, they need to be deleted from the posts before the text can be analyzed.

**Extract emojis:** these tiny little pictures are represented as quite a long HTML sequence in a post within TH, TMG, and DNMA. The html2text parser parsed emojis within the HTML code to something similar as “![:o](/Smileys/default/shocked.gif)”. As this string contains a lot of noise (real English words that were not written by the author), emojis need to be extracted in a separate step and then deleted from the text. However, emojis might also contain useful information for AA and are therefore extracted as separate features. As not all of the forums share the same emojis, 11 categories of emojis (grin, wink, smile, angry, cool, sad, roll eyes, undecided, confused, tongue, love) were defined that occur most likely in all of them. Thus, sometimes more than one emoji belongs to the same emoji category (e.g., :( and :'(
both belong to the category sad). Unfortunately, Dread does not contain animated or picturized emojis that can be identified by its HTML structure. However, its users tend to write smileys traditionally (e.g., ;), ;( ) and thus also within Dread, smileys can be extracted from posts and matched to the predefined categories.

Extracting the text: after all these steps, the pure text can be extracted from a post without any noise from quotes, PGP keys, PGP signatures, or emoji-HTML. This is done by removing all remaining HTML tags by using the package html2text version 2019.9.26.

4.2 Features

Features that are extracted from the given data are the basis for machine learning. Therefore in the following section, all features that are used within this thesis are explained. A list with all features can be found in Table C.1.

4.2.1 Lexical-based Features

Term frequency or TF-IDF features are called lexical-based features in this thesis and focus on the text that was written by an author only. Thus they contain information about the content an author writes about and are therefore also called topic-related features. However, the main focus of these features is not on the topic but rather on the frequency of words within a text. The feature extraction process of the two features listed below is implemented by using scikit-learn, version 0.22. The basic principle of the NLP technique bag of words was explained in Section 3.1.2.2 and thus will not be repeated at this point. However, for the following analysis, there are two distinct representations of the bag of words principle used.
Term Frequency: the term frequency is the most simple of the two and converts text into vectors by simply counting the frequency of words within each text. In this thesis, this feature is extracted by using the CountVectorizer of scikit-learn. The input of this algorithm will vary from character n-gram to words within different experiments to find the most suitable parameter settings.

Term Frequency Inverse Document Frequency: the second representation of BoW analyzed in this thesis is a TF-IDF approach, that multiplies the word frequency by the inverse document frequency. To extract this feature the TF-IDF Vectorizer of scikit-learn is used within this thesis. As said before, also in this case, the input of this algorithm will vary from character n-gram to words within different experiments to find the most suitable parameter settings.

4.2.2 Stylometric-based Features

The focus of this category is not on what an author is writing about, but rather how he writes a text. In the paragraphs below the features are briefly explained.

Grammar mistakes: the LanguageTool [8] is an open-source proofreading software that can differentiate between 306 distinct grammar and spelling error-groups with over 1200 distinct rules. In this thesis, the command line version of LanguageTool 4.7 is used. Each post is analyzed by this tool and all mistakes that were found are counted and stored for the later analysis.

Typos: also in this case the LanguageTool plays an important role to find and extract typos from a given text. When the message Message: possible spelling mistake found occurs within the grammar check result, then the word that was spelled wrong is extracted and also all suggestions that LanguageTool has made to correct the error. Depending on a word frequency list for the English language derived from the Brown corpus (a huge text corpus provided by the Brown University), the most
frequently used word out of the given suggestions is chosen as the most likely correct word. By comparing this word character by character with the originally written word, all differences/typos that can be found are extracted.

**Emojis:** the frequency that emojis are used differs from author to author. Therefore, the number of each emoji that appears within a post is counted and the result is added to one out of eleven categories to that the emoji belongs.

**POS Tags:** as described in Section 3.1.2.2, POS-tags are abbreviations for different types of words. The Python library nltk, which is used in this thesis, differentiates among 35 different POS-tags. To use them as features, the correct POS-tag for each word within a post is extracted first. After that, the frequency with which each tag occurs within the post is counted \((0 - \mid \text{number of words} \mid)\)

**Statistical features or style:** the remaining 59 features of this feature-category are statistical measurements of an author’s writing style e.g., the number of sentences, words, characters per word, etc. that he or she has written. For most of the basic features like the number of words per sentence, four different metrics (average, median, standard derivation, and the sum) instead of only one value are computed. A complete list of all measurements can be found in Table C.1.

### 4.2.3 Time-based Features

In previous projects [67], time-based features like the three below were under the top 10 best features for authorship attribution in the Dark Web.

**Time:** the time is split up into two features: the hour and the minute that a post was written. Hence, the time when a user is typically active within a Dark Web forum (in the morning, afternoon, during lunchtime or at night, etc.) can be concluded from these two features.
**Date:** the date that a post was posted is split up into three features: the year, the month, and last but not least the day on which a post was written. When a user was only active in e.g., 2018, then this will be noticeable.

**Weekday:** the last feature of this category is the weekday on which a post was written. This feature is neither integrated into the date nor in the time, but it might be very important e.g., to see whether some users tend to be more active during weekends and others more during weekdays. To represent the weekday in a machine-readable way, each day is transformed into a numeric value from zero to six.

### 4.2.4 Language model-based Features

The features within this category are extracted by using various language models. All of them have been introduced in Section 3.1. The preprocessing of the text before it was fit into one of the models or used to build the models themselves, included the exclusion of stop words, special characters, and words that occurred less than five times as well as a transformation of all words into lowercase.

**Sentiment:** the sentiment feature is divided into four subfeatures that specify how positive (1), negative (2), or neutral (3) a text is written. The fourth feature is a compound measurement of the other three subfeatures. All four are extracted by using the vaderSentiment package for Python version 3.2.1. VADER is the acronym for Valence Aware Dictionary and sEntiment Reasoner. It is a lexicon as well as rule-based sentiment open-source analysis tool that focuses on sentiments expressed in social media.

**Latent Dirichlet Allocation:** the probability of the occurrence of each topic within each post is calculated by LDA and used as a feature matrix. Thus, as an example, the feature matrix will have a shape of $10 \times 50$ when there are 10 topics and 50 text samples. Unfortunately, the parameter optimization of the LDA algorithm in
scikit-learn is extremely time-consuming due to a large number of iterations needed by the algorithm (multiple hours for only one out of 18 classifiers). Therefore, a trade-off between experimenting with as many parameter combinations as possible to find the optimal one, and using as little processing time as possible, had to be found. In the end, the optimization process of LDA was limited to only 30 optimization iterations instead of 100 that are used for the other features. However, even 30 is quite high compared to the default setting from scikit-learn of only 10 iterations.

**Non-Negative Matrix Factorization:** this feature/feature-matrix is extracted in the same way as the previous one except the algorithm that is used to compute the topics within the text corpus and thus the topic-probabilities per post. This is the NMF function of scikit-learn which is an implementation of the algorithm described in Section 3.1.2.3.

**Word2Vec, GloVe, and FastText:** also, in this case, the basic principle of these three algorithms was already explained in Section 3.1.2.2. An open-source version of Word2Vec, GloVe and FastText can be found in the gensim package version 3.8.1 for Python and is used within this thesis. A small test revealed that pre-trained models for Word2Vec or GloVe based on e.g., Wikipedia articles or Twitter posts were not suitable for an analysis with Dark Web forum posts. Therefore, Word2Vec, GloVe as well as FastText models were trained, based on nearly 1 million Dark Web posts that were extracted within the crawling/scraping part of this thesis. After that, the Word2Vec, GloVe, and FastText features were extracted by feeding the corresponding model with one post after another, extracting the word vectors for each word within the posts, and finally compute the average of all word vectors per post. This average vector was then used as a feature/representative of the corresponding post.
Chapter 5

Methodology of Authorship Attribution

The Authorship Attribution (AA) analysis within this thesis is very extensive, which is why the following sections are very important to be able to understand and interpret the results described in Chapter 6. At the beginning of this chapter, a brief overview of the preparation steps needed for this analysis is given. After that, the focus is switched to the setup of the analysis.

5.1 Preparations

Before the analysis could start it was crucial to find appropriate hardware (see Appendix D.1), to select authors from the dataset, to decide which classification algorithms and which preprocessing tools should be used. These points are described in more detail in the following subsections.
Table 5.1: Remaining forum combinations and number of corresponding authors after filtering out all authors with less than 50 posts in both forums.

<table>
<thead>
<tr>
<th>Name</th>
<th>Forum combination</th>
<th>Number of authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-1</td>
<td>DNMA &amp; TH</td>
<td>2</td>
</tr>
<tr>
<td>FC-2</td>
<td>DNMA &amp; TMG</td>
<td>2</td>
</tr>
<tr>
<td>FC-3</td>
<td>DNMA &amp; Dread</td>
<td>7</td>
</tr>
<tr>
<td>FC-4</td>
<td>TMG &amp; Dread</td>
<td>10</td>
</tr>
<tr>
<td>FC-5</td>
<td>TH &amp; Dread</td>
<td>17</td>
</tr>
<tr>
<td>FC-6</td>
<td>TMG &amp; TH</td>
<td>20</td>
</tr>
</tbody>
</table>

5.1.1 Selection of Authors

One of the first steps to be able to start the analysis was to find appropriate authors. In this case, the term appropriate refers to authors that can be linked via a PGP-key. On top of that, a trade-off needed to be found between the number of posts that have been written by an author within a forum (the more the better) and the total number of authors that remains for the analysis (the more the better). In a previous project [67], only authors with 500 to 600 posts were used for the analysis. Unfortunately, if this number would have been chosen in this case, only two or three authors would have remained. Thus the question was, what is a feasible number of posts for AA that keeps as many authors as possible in the analysis? To solve this question a deeper look into the dataset was necessary. In the end, the final decision was, that only authors that have written at least 50 posts in two forums were considered to be appropriate. The remaining forum combinations with more that one candidate author, as well as the number of remaining authors, are listed below in Table 5.1.

5.1.2 Preprocessing Tools

Before data is fit into a classifier it is often beneficial to preprocess it first to either fulfill the requirements of a classifier or just to improve the final results. In this thesis, three different ways of preprocessing are used and described below.
Standardization: some scikit-learn estimators are sensitive to the distribution of the data they are fitted with. They might behave badly when the data is not roughly standard normally distributed, which means Gaussian distributed with zero mean and unit variance. This is because features that have a larger variance than others might dominate the objective function of an estimator and thus make it somehow blind for other features it could learn from. Therefore, in this thesis, the StandardScaler of scikit-learn is part of the preprocessing tools and used for all those datasets that may cause the problems described above (e.g., the time-based features or stylistic features). Unfortunately, some estimators like the multinomial Naive Bayes classifier is not able to process negative data like the output of the StandardScaler. Thus, for those classifiers, the MinMaxScaler is used to scale the data of all critical datasets between two positive integers.

Normalising: normalising is especially useful when the similarity between a pair of samples should be computed. It is nothing other than scaling data to have a unit norm. Normalization is often used for text classification e.g., in the TfidfTransformer. In this thesis, the scikit-learn normalizer is used beside the StandardScaler and MinMaxScaler just to experiment with different preprocessing tools to find the most suitable one and therefore the best results.

5.1.3 Classifiers

There are several different classifiers available in scikit-learn. Some of them can be linked to the three classifier categories Support Vector Machines, Naive Bayes and Decision Trees, whose basic concepts are explained in Section 3.2.2, Section 3.2.3, and Section 3.2.4. In previous projects [67][68], all classifiers that achieved the highest results belonged to these three categories. This is why they were described in more detail. However, as there are several new features, other classifiers are used in this analysis as well. A list of all classifiers used in this analysis can be found below.
- **Support Vector Machine-based**: SVC, NuSVC, LinearSVC

- **Naive Bayes-based**: BernoulliNB, GaussianNB, MultinomialNB, CategoricalNB, ComplementNB

- **Tree-based**: DecisionTreeClassifier, ExtraTreesClassifier, RandomForestClassifier

- **Neighbors-based**: KNeighborsClassifier, RadiusNeighborsClassifier

- **Miscellaneous**: PassiveAggressiveClassifier, MLPPerceptron, Perceptron, GaussianProcessClassifier, SGDClassifier.

### 5.1.4 Final Setup of the Analysis

The analysis of this thesis is extensive because of the intention to analyze the data in as many ways as possible. Each of the six forum combinations listed in Table 5.1 is analyzed twice (once with analysis type I and once with analysis type II which are described in Section 5.1.4.1 and Section 5.1.4.2 respectively). On top of that, in both analysis types, each forum combination is once analyzed with an unbalanced dataset (with the full amount of data) and once with a balanced dataset (see Section 5.1.4.3 for more information). Furthermore, within each analysis type, each forum combination is analyzed with three versions of the dataset. Each of these three versions contains only those authors that wrote a specific minimum of posts (see Section 5.1.4.3). Until this point, there are 52 analyses (see Table D.1 and Table D.2). Besides that, within each of these 52 analyses, all four feature categories (time, lexical, stylometric, language models), as well as all features combined are analyzed with all 18 classifiers whose hyperparameters are optimized by using the RandomizedSearchCV from scikit-learn. The default parameter optimization iterations are 10, however, in this thesis, they are raised to 100 (which are 500 iterations in total due to the 5-fold cross-validation that is used), to achieve more accurate results. Furthermore, there are two different preprocessing techniques, on average, tested for each of the five feature categories and each classifier. Thus the results
described in Chapter 6 are based on $52 \times 5 \times 18 \times 2 \approx 9360$ analyses with $100(\times5)$
optimization iterations each.

The main drawback of this analysis is the time needed to compute all of these
classification analyses. On top of that, the time needed increases even further the
more samples there are, and of course, the more features to be analyzed. Thus
to control the duration of the analysis at least to some extent, each of the 9360
hyperparameter optimizations is limited to 15 minutes. The basis for the decision
was an analysis of FC-6, which took more than a week to calculate, but did not even
compute half of the results. An analysis of the log files revealed that most of the
classifiers (14 of 18) needed on average about 10 minutes per analysis. The other
four blocked the script with their calculations for several hours per analysis. At this
point, it should be mentioned that the optimization process was paralyzed with 60
threads (see Appendix D.1). However, 15 minutes was chosen as an acceptable trade-
off between computation time, completeness, as well as the quality of the results.
Due to this trade-off, not all classifiers were able to complete their computations and
are thus not part of this thesis.

5.1.4.1 Analysis Type I: Combined Analysis

Most common for AA is to extract a text corpus that includes all texts from all
authors and split this corpus into a training set and testing set. This is done in the
first part (type I) of the analysis by putting all posts from both forums of each forum
combination listed in Table 5.1 into a single dataset. After that, the posts are mixed
and split up into 70% training data and 30% testing data. The advantage of this type
of analysis is that all posts from all linked authors can be used. However, a major
disadvantage is that the proportion of posts from forum A and B vary significantly
from author to author. Therefore, the main focus of this analysis is on the feasibility
of AA, based on posts from different Dark Web sources. The results of this analysis
type will always be visualized in blueish colors in all the diagrams shown below.

5.1.4.2 Analysis Type II: Separate Analysis

The second part of the analysis is based on training sets that contain only posts from forum A of a given forum combination and testing sets that contain only posts from forum B of the same forum combination. The differences between analysis type I and II are visualized in Fig. 5.1. The forum that has more posts per author on average is used for the training set and the other for the testing set respectively. Furthermore, the proportion between the two sets remains the same as in type I (70%/30%), which unfortunately leads to a high loss of data. To make this problem more apparent it is explained with an example. Imagine the posts from forum A are used for training and those from forum B for testing. An author wrote 1000 posts in forum A but only the minimum of 50 posts in forum B. Hence his 50 posts are the maximum that can be used for testing and it is known that the testing set always contains 30% of the total amount of data from each author. Thus, the question is, how many posts from 1000 remain for training when 50 posts are 30% and the training set should contain 70%? The simple percentage calculation needed to solve this question can
Fig. 5.2. Each bar shows the sum of all posts from all authors within the corresponding forum combination and type of analysis.

be found in Eq. (5.1). The proof of the calculation can be found in Eq. (5.2).

\[
\frac{|post_{\text{test}}|}{30} \times 70 = |post_{\text{train}}| \Rightarrow \frac{50}{30} \times 70 = 116.6
\]  

(5.1)

Proof: \(116.6 + 50 = 166.6 = 100\% \Rightarrow \frac{166.6}{100} = \frac{5}{3} = 1\% \Rightarrow \frac{5}{3} \times 30 = 50 = 30\% \)  

(5.2)

The result shows, that only 117 posts (the number of posts is always rounded up) from 1000 posts can be used for training. The same problem can occur vice versa in the testing set. However, when it occurs, then only the longest posts written by the concerned author were chosen for the corresponding data set. The extent of this problem is illustrated in Fig. 5.2. The main disadvantage of this type of analysis should now be clear, however, there are also positive aspects that should be mentioned at this point. This type of analysis can reveal which features can achieve good results even when the author might have changed some of his typical behaviors between the training and testing forum. Therefore the focus of this analysis is on the suitability of the extracted features. The results of this analysis type will always be visualized in greenish colors in all the diagrams shown below.
Fig. 5.3. The boxplots show the unbalanced distribution of the number of post per author within each forum combination and for each analysis type.

5.1.4.3 Sub-analyses

Both analyses (type I and II) are further divided into several “sub-analyses” that are listed below. A complete list of all analyses can be found in Table D.1 and Table D.2.

Number of posts per author: as already stated in Section 2.2, the more text from an author that exists, the better is the probability of a successful AA analysis. Thus, there are three different sub-analyses for each analysis type. In the case of analysis type I, this is an analysis with all authors, one with only those authors that wrote more than 500 posts, and the last with only those authors that wrote more than 1000 posts. As the total number of posts per author is lower in analysis type II, one sub-analysis is based on all authors, the second on all authors that wrote more than 200 posts (summed up over both forums), and the last on authors that wrote more than 400 posts. The aim of this sub-analysis is to figure out how many Dark Web forum posts are needed per author for an AA analysis.
**Balanced and unbalanced:** a major problem of both analysis types as well as the previously mentioned sub-analyses is that some authors wrote a huge number of posts whereas others just wrote only a few hundred or fewer (see Fig. 5.3). Therefore, to analyze the effect of this problem on an AA analysis, all datasets are analyzed twice; one time unbalanced and the other time balanced. Balanced means that the number of posts of all authors is limited to the number of posts written by the author with the fewest posts within the dataset. Similar to the procedure in analysis type II, only the largest posts are kept when shrinking the number of posts of an author. In the case of analysis type I, also the proportion of posts from forum A and B was balanced as much as possible to push this type of analysis more in the direction of the main research question of this thesis, which was AA of posts written in two different Dark Web forums.
Chapter 6

Results

The results described in this section are selected results for each forum combination and each feature category within each analysis mentioned in Table D.1 and Table D.2. In this thesis, the best results of an analysis and/or classifier are considered to be those with the highest F1-score and the highest accuracy. This is based on the fact that a high recall, as well as a high precision, are essential for AA in the Dark Web to assign a post to its correct author as reliably as possible. A high precision indicates that posts from other authors are relatively rarely assigned to a considered author X. A high recall, on the other hand, indicates that many posts written by author X have been correctly assigned to author X. Since the F1 score is a kind of average between precision and recall (see Section 3.2.1.3), it is an appropriate evaluation criterion. The accuracy is used to rank the results but is not presented in the following diagrams as it was always very close to the F1-score and would overload the charts.

When focusing at Fig. 6.1 and Fig. 6.2 it can be observed, that AA within this analysis becomes more difficult the more candidate authors there are. It is possible to achieve an F1-Score of nearly 100% by focusing only on six time-based features and considering only two authors, whereas a similar result within forum combinations
Fig. 6.1. Overview of the F1-Score achieved within the analyses of posts from all authors of all six forum combinations (FC-1 to FC-6) for each feature category with **analysis type I** by using either an unbalanced dataset (left) or balanced dataset (right). #A denotes the number of authors within the corresponding forum combination and #P the number of posts.

with a larger number of authors cannot be observed. Furthermore, the balanced analyses appears to increase the results in most cases. However, there is a significant difference between the results of analysis type I and analysis type II, which is most significant for datasets with more than two authors (FC-3 to FC-6). The results of the green (type II) analysis are, in general, significantly worse (10% to 20%) than those achieved within the blue (type I) analyses. Even when Fig. 6.1 and Fig. 6.2 show only the results of the balanced and unbalanced analyses that consider all authors, they show a tendency that can also be found in the remaining analyses (see Fig. D.1 to Fig. D.4).

The results achieved by the different feature-categories differ significantly among each other and between the different types of analyses as shown in Fig. 6.1 and Fig. 6.2. Therefore, in the following sections strengths and weaknesses of the different categories that can be concluded from the given results of FC-5 and FC-6 are described. These two forum combinations are chosen for this more detailed analysis because they represent the problems and difficulties for a successful AA analysis. It
Fig. 6.2. Overview of the F1-Score achieved within the analyses of posts from all authors of all six forum combinations (FC-1 to FC-6) for each feature category with analysis type II by using either an unbalanced dataset (left) or balanced dataset (right). # A denotes the number of authors within the corresponding forum combination and # P the number of posts.

should be noted that paragraphs within the following section always have the same structure. At first, the detailed results achieved by the given feature category are shown and briefly described considering not only posts that are correctly attributed to its author but also the likelihood that the correct author is within the top three highest ranked candidates. Within the area of the Dark Web, it is also interesting to know if the correct author of a post would have been within the three highest-ranked candidates. For law enforcement agencies it would be desirable to attribute most of the posts correctly, for sure, but it would also help if the group of highly probable suspects could be reduced to just three people. On the other side, this would also be a dramatic danger for the anonymity of Dark Web users who try to hide within a crowd. However, this is followed by a detailed description of characteristics from authors that could be classified best by using the corresponding feature category. After that, all those authors are analyzed that were not easy to identify. Both of these paragraphs focus on the results of analysis type I (blue) only, since this analysis was based on more data and thus it is considered to be more reliable. However,
the last paragraph of each of the following sections focuses on the suitability of the corresponding feature category for analysis type II (green).

### 6.1 Time-based Features

The time-based features belong to the most accurate features of all within both analysis types of FC-1 and FC-2 (2 authors in both cases). However, for the other four forum combinations FC-3 (7 authors), FC-4 (10 authors), FC-5 (17 authors), and FC-6 (20 authors), it drops significantly from an F1-score of 99% (analysis type I) to only around 75% (see Fig. 6.1). For analysis type II the situation is even worse (see Fig. 6.2). However, when taking a closer look at Fig. 6.3 it can be seen that the correct author of a post is 88% (FC-6) and 91% (FC-5) within the three most likely candidates when analyzing the corresponding dataset with all authors and type I.

Except the results themselves, the analysis of the classifiers that achieved the best results within all 52 analyses listed in Table D.1 and Table D.2 shows a significant trend for the time-based features. Within 22 analyses, the ExtraTreesClassifier achieved the best results of all tested classifiers and in 18 cases, it was the Ran-
domForestClassifier (see Table D.3). Therefore, it appears that tree-based classifiers achieve the best results when time-based features are analyzed stand-alone.

**Which authors can be identified best?** There are four authors in FC-5 (A_9, A_10, A_11, A_12) and four authors in FC-6 (A_0, A_7, A_8, A_12) that achieved better results than all others within the unbalanced analysis type I and which are the focus here (see Fig. 6.4). Within FC-5, one main reason for the good performance of the time-based features is the number of posts per author. There are only six authors that wrote over 500 posts and four of them do belong to the best recognizable authors. The remaining two are also identified quite well but not as good as the four others.

In contrast, the results of FC-6 show that writing many posts is one reason for being identified quite well with the time-based features, but they also show that it does not necessarily mean that an author is unidentifiable when they have written only a few posts. In this forum combination, three out of the four best identifiable authors wrote more than 1000 posts whereas the remaining one achieved comparatively equivalent results with only about 350 posts. This indicates two facts: in general, the more candidate authors there are, the higher the number of posts per author is required so that an author can be identified using time-based features. On the other side, some authors do not need a high number of posts because their daily rhythm seems to be so atypical that they stand out easily. The atypical thing on A_8 is that they appear to be a Dark Web newbie or at least a TH and TMG newbie. They started writing posts in August 2019 until the end of the crawl in November and wrote the most posts between the 11th to 14th as well as between the 19th and 23rd day of a month.

When taking a look at the balanced analysis that considers all authors, it is easy to see that the F1-score of those authors that wrote many posts (all four within FC-5 and three out of four within FC-6) drops significantly. In contrast, the same score of the remaining author with a smaller number of posts increases by 4%. The reason
Fig. 6.4. The detailed results of all authors within the balanced and unbalanced analysis (type I) of FC-5 (left) and FC-6 (right) by using **time-based features** only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).

For this is quite obvious, as only 104 (FC-5) or 140 (FC-6) posts per author remained in the dataset, which seems far too few for good results with the time-based features. This overall tendency also remains when considering only authors with over 500 or even 1000 posts (see Fig. [D.1](#) and Fig. [D.2](#)).

When balancing the dataset by choosing posts at random and not the longest ones, it can be seen whether the length of posts plays an important role in a feature. The time-based features themselves are not influenced by the length of a post since each post does only have exactly one timestamp. Although, one might expect that authors tend to write longer posts at a different time than smaller ones because they might not always have the time for writing a long text. However, the overall F1-score is nearly identical (1% more or less) between the balanced dataset with only long posts and that with randomly chosen posts. This leads to the conclusion that there is, in general, no connection between the length of a post and the time when a post is written.

Another characteristic of the time-based features was found when reducing the number of candidate authors. This procedure leads to an increase in the result of the best
authors within analysis type I, which was already visible when comparing the results of the smaller datasets FC-1 to FC-3 with those of FC-5 and FC-6. The reason for this is easily explained. Humans tend to have slightly different daily rhythms. This means that everyone has their own time to get up, go to work or school, have leisure time, or go to bed. M. La Morgia et al. have shown that when the activity of Twitter users throughout the day is graphically illustrated, the shapes of the resulting graphs are remarkably similar (see [50]). When there is a difference, then the shapes are mostly just shifted to the left or right on the time axis. In the case of their research, this was mainly due to the timezone a user lives in. Thus when only a few authors are considered, then their daily rhythm might be differentiated more easily due to their specific habits or due to the timezone they live in (see Fig. 6.5). On the other side, when there are many authors then their rhythms rather tend to overlap each other rather than to expose them (see Fig. 6.6).

Which authors cannot be easily identified? Within FC-5, seven authors can just be recognized very poorly and six within FC-6. As previously stated, balancing the dataset reduces the F1-score of the well-identifiable authors. Interestingly, the complementary effect can be seen when focusing only on poorly identifiable authors. The F1-score of all seven of FC-5 and three authors out of six within FC-6 increases due to this procedure.
The daily activity of all users that are considered within FC-6 tend to overlap each other and thus make it difficult to identify a specific author by focusing on time-based features only.

The number of posts is crucial for being able to identify an author in FC-5 correctly. However, three out of the six authors from FC-6 have written more than 500 posts and are still comparatively difficult to classify. As none of the six poorly identifiable authors within FC-5 wrote enough posts to also remain in the datasets with a reduced number of authors, it is interesting to see what happens with the results of the three authors of FC-6, who have made the leap into one of these datasets.

A deeper look into the analyses with only those authors that wrote at least 500 posts reveals, that the F1-score of these three authors remains similar in the unbalanced analysis but increases significantly (by at least 10%) when the dataset is balanced. This indicates that a balanced dataset is generally essential for those authors that are not that easy to identify by using time-based features only. In contrast, the post length does not seem to influence the results of any of the 13 worst identifiable authors for the same reasons mentioned.

**Suitability for analysis type II:** as already stated at the beginning, the time-based features do not seem to work well within analysis type II when considering more than two authors. Interestingly, the authors that are identified best, and those that are recognized worst, have mostly the same properties as in analysis type I (see Fig. [D.5]). Therefore, the most obvious reason for the poor results of the time-based features within analysis type two would be, that there are simply not enough posts
for each author to be able to find all daily rhythms. On top of this problem, the worst identifiable authors have one main commonality (at least in FC-6): they were not active in both forums in the same years. Unfortunately, when deleting the year from the feature set due to this observation, the overall F1-score decreases. This happens because the results of the previously well recognizable authors decrease due to the loss of the year.

As the results within analysis type I are comparatively high, it seems that authors tend (at least) to be active to a similar time within their favorite forum in that they have written the most posts. This assumption is based on the fact that the proportion of posts from forum A and B within the dataset of analysis type I is highly biased (for some classes) by content from either the one or other forum. As there are too few posts within analysis type II to find the daily rhythms, it remains an open question of whether authors tend to be active at the same time within both forums. However, the results of the analysis type I show that there is at least a tendency in that direction.

6.2 Stylometric-based Features

Compared to the time-based features, the results of the stylometric-based features are worse within analysis type I whereas they are, in general, better within analysis type II (see Fig. 6.7). The probability that a post can be correctly attributed to its author ranges around 40% in the unbalanced case and up to 60% in the balanced case within FC-5 and FC-6. Compared to that, the probability that the correct author can be found within the top 3 highest ranked candidates ranges between 50% and 83% in these forum combinations.

Even in the case of the stylometric-based features, a few classifiers were able to achieve the best results in the vast majority of analyses. (see Table D.3). Within
Fig. 6.7. Comparison of the probability that a post is classified to its correct author (top 1) or that the correct author is within the three most likely candidates (top 1-3) when considering all authors and **stylometric-based features** only.

26 of 52 analyses, the LinearSVC achieved the best results of all tested classifiers followed by the ExtraTreesClassifier (11 times) and the MLP Classifier (9 times). The only type of analysis where the LinearSVC does not dominate the results columns of the stylometric-based features is for FC-6 within analysis type I. In that single case the ExtraTreesClassifier seems to achieve the best results.

**Which authors can be identified best?** When FC-5 and FC-6 are analyzed with analysis type I, the factors that lead to a high F1 score by using stylometric-based features only are much clearer than in analysis type II. The most important is the number of posts (see Fig. 6.8). Seven out of the eight best authors (FC-5: A_9, A_11, A_12), (FC-6: A_7, A_10, A_12, A_18, A_19) have written more posts than the average. The number of posts written by the remaining author is 3.5% fewer than the average and thus is very close to the mean. However, the average number of posts remaining just for testing for these authors is 475, which is comparatively high. Furthermore, this might be one of the main reasons why the F1-score of authors that could be identified best are more negatively affected when the number of posts is reduced to balance the dataset. The score of six out of eight authors drops by 2% to
21% when balancing the dataset, which might be due to the average length of the posts that remain. There are two possible reasons. Either the typical writing style of these authors can be found in comparatively short posts, or the number of posts is too low to be able to find a typical writing style for these authors. The justifiability of the latter can be seen when choosing the post at random when balancing the dataset. In that case, the decrease of the F1-score of all eight authors considered here ranges between -5% to -26%, which is similar to the other balancing strategy. This leads to the conclusion that the stylistic pattern that makes these authors expose most, can only be found when analyzing a great number of posts (and thus a mixture of long and short posts).

When taking a look at the results from the two analyses with authors that wrote at least 500 or 1000 posts, then it can be seen that all eight authors remain in the second-largest dataset (which contains authors with at least 500 posts) (see Fig. D.10 and Fig. D.11 respectively). Furthermore, five out of these eight also remain in the largest one. This fact was to be expected as all eight authors wrote many posts compared to the rest. Interestingly, the impact of the balancing process on the good authors changes when the average number of posts per author increases. When focusing on the balanced analysis of authors that wrote at least 500 posts, then the F1-score of six out of eight authors rises due to this process and does not decline as when considering all authors. Furthermore, when the balanced dataset contains only authors with 1000 or more posts, the F1-score of all of the remaining five authors increases. The most likely explanation for this is that the more posts remain per author when balancing the datasets, the more individual stylistic patterns can be observed and thus can be used to better differentiate the authors from each other. However, there is one main commonality to the results from the analysis of the balanced dataset including all authors. The F1-score drops slightly when choosing the posts that remain in the balanced dataset at random and not by focusing on the
Fig. 6.8. The detailed results of all authors within the balanced and unbalanced analysis (type I) of FC-5 (left) and FC-6 (right) by using **stylometric-based features** only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).

The authors are sorted by the number of posts from the left (few posts) to the right (many posts).

**Which authors cannot be easily identified?** When using stylometric-based features only within an analysis of type I of FC-5 and FC-6, 19 authors are considered as being poorly detectable. One significant commonality is that they have written considerably fewer posts (≈ up to -64%) than the average. On top of that, the average number of posts remaining within the testing set for these authors is 82 posts per author, which is comparatively few. Interestingly, when the total number of posts per author is reduced to the minimum (when balancing the dataset), then the F1-score for 18 out of these 19 authors increases by at least 7% and up to 44%. One reason for this might be the average length of the posts, which rises by 200 words per post within FC-6 and even 250 words per post within FC-5. Since poorly detectable authors did not write that many posts, the longest ones seem to contain...
more stylistic features that make authors more easily to detect. Another fact that influences the results is the number of posts per author within the balanced dataset. When choosing posts at random and not the longest ones during the balancing process, the tendency that poorly detectable authors are recognized better remains. Although, the effect is not as great as before since the F1-score increases in only 13 and not 18 out of 19 cases and only from 5% up to 26% and not up to over 40% like in the previous case. However, it is significant enough to conclude, that balancing the dataset makes the poorly identifiable authors stand out better. Thus, the conclusion is, that balancing increases the results of these authors but decreases those from the best identifiable ones. Therefore, a tradeoff needs to be found that keeps as many posts as possible for each author from the latter class within the dataset, but at the same time, reduces the number of posts as much as possible so that the others have a chance to stand out.

**Suitability for analysis type II:** the overall results of analysis type II with stylometric-based features only are significantly (around 20%) lower than those within analysis type I. However, they are better when focusing on time-based features only. A detailed analysis reveals that there are several reasons for this difference. The first is that the stylometric-based features seem to have, in general, a hard time dealing with authors that wrote lots of very small posts (see Fig. D.6). However, in analysis type II there is one exception, which is an author (A_16 in FC-5) that wrote comparatively few posts and also comparatively few words per post but with an extremely low standard deviation between the training and testing set (see Fig. D.12 and Fig. D.13 for more details). Therefore it seems that an author can be identified well even when they write small posts within analysis type II, but only if they write about the same number of words per post within both forums. A deeper look into the data revealed that the author typically wrote posts with a small paragraph of two to five rows at the beginning, followed by an empty line and a greeting or
a short signature. However, this author is an exception among authors who have written comparably few and small posts. Interestingly, there is a tendency that can be observed about the distribution of these small posts. Most of the authors have written them only within either the training set or testing set which means either one or the other forum. This results in a significant difference in the post length between training and test set (also visible in Fig. D.14 and Fig. D.15 for FC-6) which is difficult to cope by the stylometric-based features as e.g., the post length, is one of the features that are included in the stylometric-based features of Section 4.2.2. Furthermore, the balancing process helps only in the case when the dataset is reduced to the longest posts of each author. In that case, the F1-score of most of the worst identifiable authors increases. Except that, the common ground of authors that achieved the best results within analysis type II is a long post length on average. When balancing the dataset to the longest posts, their F1-score further increases even when the number of posts drops significantly. In contrast, the F1-score decreases when choosing the posts at random, which is another indicator that longer posts make authors more easily detected within analysis type II. However, most of the authors that achieved comparatively poor results within analysis type I do so in analysis type II as well. When they do not belong to the worst identifiable authors, then the results show that they have most likely written more posts than the average within analysis type II.

Therefore, the overall conclusion when considering the results of analysis type I and II is that authors seem to write more passionately in either one or the other forum, which results either in a different style of their posts or in a different number of posts per forum that are available for the analysis. Both cases are a problem for the stylometric-based features especially within analysis type II.
6.3 Lexical-based Features

The results shown in Fig. 6.9 prove that the lexical-based features belong to the best features within this analysis. When an unbalanced dataset is used, then the probability that the correct author of a post is within the top three candidates is above 60% or at least very close to 60% in nearly all cases (analysis type I and type II). On top of that, when the analysis is based on a balanced dataset, the probability to find the correct author within the top three rises to at least 66% and up to 86%. When taking a closer look at the best classifiers listed in Table D.3, it is obvious that Naive Bayes classifiers seem by far the most suitable type of classifiers when analyzing the lexical-based features. In 19 of 52 cases, the ComplementNB classifier achieved the best results and in 18 of the remaining 33 analyses, it was the MultinomialNB.

**Which authors can be identified best?** There are five authors within FC-5 (A_3, A_6, A_9, A_10, A_15) and seven in FC-6 (A_6, A_7, A_9, A_10, A_12, A_18, A_19) that achieved comparatively high F1-scores (see Fig. 6.10). When balancing the dataset with regard to the longest posts, the F1-scores from 7 out of these 12 authors increases by at least 1% but up to 20%, for 2 it remains the same and for
the remaining three authors it decreases by 5 to 15%. When balancing the dataset by choosing posts at random, the scores for 8 out of 12 authors decreases by up to -25% and around -15% on average, it stays the same for one author and increases only for the last remaining author (A_6 in FC-6). This author is characterized by the longest posts on average within the whole dataset. They wrote 750 words on average within each post with a high standard deviation of over 1500 words but also by far the highest median of all authors with 400 words (see Fig. D.11). The other 11 authors can be categorized into two groups, those that belong to the authors that have written the most posts (9 from 11 see Fig. 6.10) and the remaining two authors that wrote comparatively long posts (see A_6 and A_15 in Fig. D.10). Interestingly, all well-identifiable authors who have reduced or not improved their F1-score during balancing (FC-5: A_3, A_9, A_10; FC-6: A_10, A_19) belong to the category of authors with many posts. When taking a look at their post lengths, then a typical pattern can be observed. All of them wrote many, but small posts on average (see Fig. D.10 and Fig. D.11). When reducing the dataset to only a fraction of these, and on top choosing only the longest post, which is obviously not the type of post these authors usually write, then the classification results decrease. Thus, an author that wrote long posts stands out from the crowd more easily when considering only a few posts, than an author that usually writes many short posts. This means that the length of the posts is the most important influencing factor for the lexical-based features. Authors who have written many short posts only seem to be among the best identifiable authors if the classifier is provided with all of their posts.

When reducing the number of candidate authors per dataset to only those that wrote 500 posts or more, then the F1-score of 7 of the 9 remaining well-identifiable authors increases by 1% to 10%. However, when comparing the overall F1-scores of the unbalanced dataset when considering only authors with at least 500 posts, and the balanced dataset including all authors, then it becomes clear that focusing on
a few longer posts per author is, in general, more or at least equally effective than shrinking the number of candidate authors. This tendency continues also between the “balanced 500 +” dataset and the unbalanced dataset with only those authors that wrote at least 1000 posts (see Fig. \ref{fig:D.1} and Fig. \ref{fig:D.2}).

**Which authors cannot be easily identified?** When focusing on lexical-based features only, six authors from FC-6 and seven from FC-5 are difficult to detect. For 11 out of these 13 authors the reason can be found quickly when taking a look at the results of the two different balanced analyses. The F1-score of all of them increases significantly around 25% on average, but by at least 7% and up to over 40% when analyzing only their longest posts. Thus, the focus on the longest posts seems to be very promising. However, for ten of these authors, the F1-score also increases when the dataset is balanced with posts chosen at random, but only by around 15% on average. This leads to the conclusion that the balancing process makes at least 10 out of the 13 authors considered here stand out more clearly. This is not a surprise as they all share the same characteristic: they have written comparatively few posts.
Only two authors are an exception to this rule, which have not been included in these results. One of them has written over 500 posts, the other over 1000. When balancing the dataset to only the longest posts, their results increase by around 50% and 8% respectively. Whereas the results of the former-mentioned author increase by only 9% instead of around 50% within the balanced analysis, where the remaining posts are chosen at random. On top of that, the results of the other author even decrease by around 20%. The reason can be found in their post lengths. Both of them have a very high standard deviation of around 700 words per post and they rank 1 and 3 of the authors with the fewest words per post on average (see A.0 and A.12 in Fig. D.10). Furthermore, they also have the smallest median for the number of words per posts. Therefore, to be able to increase the scores of poorly identifiable authors, but at the same time keep the scores of those that can be identified best as high as possible, the dataset needs to be balanced by choosing the longest posts of each author.

**Suitability for analysis type II:** in general, the lexical-based features are suited comparatively well for authorship attribution within analysis type II. Furthermore, the statements made before about good identifiable and poorly identifiable authors remain similar also in analysis type II (see Fig. D.7). An author can be identified well when they either tend to write long posts or when they have written small but many posts. Focussing on the longest posts only is also essential in this analysis. This can be concluded when comparing the results of the balanced analyses which are putting the longest posts in the spotlight and the corresponding unbalanced analyses. Except for one case, the unbalanced analyses are not able to achieve as promising results as the balanced analyses. Another proof for this statement is that the F1-score of too many authors decreases when choosing the posts at random. As the overall score of the lexical-based features within analysis type II is better than that of the stylometric-based features, it seems like authors rather tend to
write about the same things or at least use similar words within two different forums than to use the same writing style when posting a post.

### 6.4 Language model-based Features

The language model-based features, as they are extracted so far, are not suitable for AA within the Dark Web (see Fig. 6.11). Especially datasets with a large number of authors (FC-6) seem to be problematic for these features, as in the best case the correct author of only 62% of all posts is listed within the top 3 most likely candidates. FC-5 has three authors less than FC-6, which seems to lead to a slightly better probability to find the correct author within the top 3 candidates (up to 75%). When taking a look at Table D.3 it is obvious that the LinearSVC (25/52) and the PassiveAggressiveClassifier(13/52) seem to achieve the best results when analyzing the language model-based features only.

**Which authors can be identified best?** There are a few authors that can be recognized better within FC-5 (A_2, A_3, A_5, A_6, A_9, A_10, A_12) and FC-6 (A_6,
Fig. 6.12. The detailed results of all authors within the balanced and unbalanced analysis (type I) of FC-5 (left) and FC-6 (right) by using language model-based features only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).

A_9, A_10, A_12, A_18) but there is no single common reason which is characteristic for all authors (see Fig. 6.12). Some have many posts and others do not, the same applies to the length of the posts, some have long posts, others not (see Fig. D.10 and Fig. D.11). However, even when there is no clear trend for all authors that were classified best in the very first analysis (unbalanced, all authors considered), there is at least one fact that can be observed. When an author has written many posts (500 or more) then it is more likely that this author can be identified comparatively well. For both forum combinations FC-5 and FC-6 only three authors that were categorized as being not easy to identify wrote more than this limit.

**Which authors cannot be easily identified?** As previously stated, it is extremely difficult to explain why an author achieved good or bad results when focusing on language model-based features only. However, poorly recognizable authors seem to have two characteristics in common. When an author has written just a few posts, it is likely that they are not as easy to classify. The second characteris-
tic in common is the average length of the posts. When an author tends to write small posts, then it is more likely that a classifier cannot classify them by using only language model-based features. Unfortunately, this does not seem to apply to everyone, as some authors with small posts belong to the best recognizable authors and some with long posts to the poorly identifiable authors. However, the proportion of authors to that this general observation does not apply (in either one or the other direction) ranges around 30%.

**Suitability for analysis type II:** the language model-based features appears to be neither suitable for an analysis with many authors, nor for analysis type II (see Fig. [D.8]). The best-achieved results are an FC-1 score of 55% when reducing the number of authors within FC-6 to only 6 (only those with 1000 or more posts) and when balancing the dataset. However, the few tendencies that were already visible in analysis type I for well-identifiable or poorly-identifiable authors can also be observed in analysis type II. This leads to the conclusion that either most authors within the testing and training set are writing about the same topic and use words with a similar meaning, or authors within Dark Web forums do not tend to write about the same topic within two different forums at all. The former assumption is more likely, as most of the posts are about drugs, drug trading, fraud or cryptocurrency.

### 6.5 All Features Combined

When putting all features together and analyzing them combined, one might expect that the results must increase because all information is now merged. However, the real-life experience shows, that this is not the case within most of the analyses in this thesis. In Fig. [6.13] it is noticeable, that the probability that a post can be correctly attributed to its author when focusing on all features combined is comparatively low (40% or less). However, there is a clear tendency that this probability increases when
Fig. 6.13. Comparison of the probability that a post is classified to its correct author (top 1) or that the correct author is within the three most likely candidates (top 1-3) when considering all authors and all features combined.

balancing the dataset. In the best case, it rises to 80% and even higher (to 86%) when ranks two to three are included.

As visible in Table D.3 the best classifiers for the analysis with all features combined seem to be the PassiveAggressiveClassifier (20/52), the LinearSVC (12/52), and the ExtraTreesClassifier (9/52).

**Which authors can be identified best?** In contrast to the language-model based features in the section before, some typical characteristics for the best classifiable authors are visible when combining all features within analysis type I. Eight out of eleven authors (FC-5: A_{10}, A_{11}, A_{12}, A_{15}, A_{16}; FC-6: A_{3}, A_{6}, A_{10}, A_{12}, A_{18}, A_{19}) that can be classified best have written more than 500 posts and from the remaining nine authors with many posts, seven are at least moderately well classifiable (see Fig. 6.14). When balancing the dataset so that only the longest posts remain, a significant increase in the F1-score for 10 out of 11 of the best identifiable authors by at least 12% and up to 46% with an average around 25% is visible. The confusion matrices in Fig. D.17 to Fig. D.19 show this trend in more detail. Also when balancing the dataset by choosing posts at random, the F1-score of 7 out
of these 11 authors increases by 3% to 46% with an average of 20%. This shows that there is a tendency that balancing posts is a little bit more important than the lengths of the posts. In any case, it is more effective to balance the complete data set than to reduce the number of authors to only those who have written at least 500 posts or more (see Fig. [D.1]). This tendency is quite similar for the step from a balanced data set with 500 posts per author to an unbalanced data set with even fewer authors (those with more than 1000 posts) (see Fig. [D.2]).

**Which authors cannot be easily identified?** There are 15 authors categorized to be poorly classifiable within FC-6 and FC-5 when focusing on all features combined within analysis type I. Twelve of these have written comparatively few posts, which is a significant characteristic. However, when balancing the dataset by choosing only the longest posts, the F1-score of all 15 authors increases dramatically by at least 19% and up to 59% and with an average of 38%. Five of these authors even exceeded the results of the authors who were among the best without balancing. In the other balanced analysis when posts are chosen at random, the F1-score of all 15 authors
increases by 25% on average, which is a clear indication that balancing the dataset leads to an increase of the overall performance when analyzing all features combined.

**Suitability for analysis type II:** analyzing all features combined within analysis type II does not have the positive effect that one might expect. Balancing the dataset, which leads to a high increase of the F1-score within analysis type I does not have the same significant effect within analysis type II, even when a positive tendency is visible (see Fig. D.9 compared to Fig. 6.14). This is because poorly-identifiable authors get a higher F1-score at the expense of the good authors, whose F1-score decreases on average when the dataset is balanced (for more details see Fig. D.9 or the confusion matrices in Fig. D.20 to Fig. D.23). Reducing the number of authors does improve the overall result within FC-5 but not significantly within FC-6.

In contrast to analysis type I, a high number of posts does not necessarily indicate that an author is easily classifiable. Five authors with more than 200 posts were categorized to be poorly identifiable whereas four authors with a similar number of posts belong to the best identifiable ones. Unfortunately, a reason for this cannot be found as the F1-score of some authors with many large posts decreases whereas it increases for others. Even in the opposite case, if an author has written many short posts, they will achieve a higher F1-score in some cases and a lower one in others. Thus, a unique pattern cannot be found without investigating each post individually.

### 6.6 Voting

The results described in the previous three paragraphs are somehow disillusioning as the combination of all features is not as good as some feature categories alone. The question is, where did the potential of the individual features get lost in the joint analysis? The answer is surprisingly simple when taking a look at Table D.3. Each feature category can be classified best by different classifiers. Thus, when putting all
features together and classifying them with only one classifier, the results decrease. Therefore, another approach to analyzing all features combined was tested at the very end of this thesis. In contrast to the previous one, the features are not put together in one single dataset. Instead, they are classified as stand-alone by the same classifier with the same classifier parameters as before. However, this time, the result of each classifier is fed into a final voting classifier. Thus, this final classifier receives four results from four different classifiers for each sample that is used for testing. Out of these results, the voting classifier computes the most likely candidate author for each post. Since it is known from the previous analyses when each feature category works well, weights can be added to the computation so that those classifiers that are more reliable in a given situation than others, have more influence on the final result. E.g., when focusing on analysis type I with an unbalanced dataset, it is known that the time-based features work very well, the stylometric-based and lexical-based features are not as suitable as the time-based features but still work well, whereas the language model-based features achieved the worst results. This knowledge produces a tendency for which features should be weighted more or less. However, the final weights still need to be determined by experimenting. In the case
of the previous example, the weights that lead to the highest results of the voting classifier were: time-based (6), stylometric-based (2.5), lexical-based (2.5), language model-based (1). Mathematically, the computations made by the voting classifier can be described as in (6.1) and (6.2) where $B \in \mathbb{R}^{n \times j \times k}$ denotes a three-dimensional matrix that contains the probability estimates of all classifiers (the probability of the posts for each author in each model) and $A \in \mathbb{R}^{j \times k}$ denotes a matrix that contains the weighted averages $a_{m,l}$ of the probability estimates.

$$a_{m,l} = \frac{\sum_{i=0}^{n} b_{i,m,l} \cdot w_i}{\sum_{i=0}^{n} w_i} \forall a_{m,l} \in [0, 1] \quad (6.1)$$

for $m \in \{0, ..., j\}$, where $j$ denotes the total number of samples, for $l \in \{0, ..., k\}$, where $k$ denotes the total number of authors, for $b_{i,m,l} \in B_{n \times j \times k}$, where $n$ denotes the total number of classifiers, and where the vector $w$ contains the weights for each classifier. The final output of the voting classifier can be mathematically described as shown in (6.2):

$$v_m = \arg \max_{l \in \{0, ..., k\}} a_{m,l} \quad (6.2)$$

where $\arg \max$ computes the column index of the author/column with the highest probability and thus the most likely author within each row (sample) of $A$.

Fig. 6.15 shows the significant increase of the results achieved by voting all feature categories instead of putting them all together into one single dataset and analyzing them by a single classifier (see Fig. 6.13).

Especially within analysis type I the results of the previously unsuitable unbalanced datasets increases to 80% in the case of FC-6 which is the largest forum combination with 20 authors. A similar trend can be seen for FC-5, which achieves an F1-Score of 87% when voting the results of all feature categories. Interestingly, the results of analysis type II, as well as those of the balanced analyses, do not increase that much.
Fig. 6.16. Detailed overview of all results (incl. subfeatures) achieved by analysis type I (balanced) when analyzing FC-5 and FC-6. The graph shows, that subfeatures that are analyzed separately do not achieve better results than when they would have been analyzed combined with the remaining features of their corresponding feature category. On top of that, the last four columns on the left side show that voting (no matter which version) achieves better results than the analysis of all features with a single classifier. As in previous charts, #A denotes the total number of authors within a dataset and #P denotes the total number of posts.

There might be multiple reasons for that. All datasets of analysis type II as well as the balanced datasets of analysis type I contain significantly fewer posts. This leads to an increase of the F1-score of the lexical-based features but to a decrease of the score for the time-based features. As the latter ones did have the most influence on the voting results of the unbalanced datasets, it is not surprising that the results of the balanced datasets do not increase in the same way. In the case of analysis type II, there were only a few feature categories that passed the 50% limit at all. Thus, the voting classifier is not able to improve the results of this final analysis when most of the feature categories are not able to pass the 50% limit stand alone.

In Fig. 6.16 and Fig. 6.17 a more detailed overview of the results achieved by all features and different types of voting analyses is shown. “Voting all Features Separately” denotes a voting analysis of all subfeatures without the results of the corresponding feature category where all subfeatures are combined and analyzed with a
Fig. 6.17. Detailed overview of all results (incl. subfeatures) achieved by analysis type II (balanced), when analyzing FC-5 and FC-6. The graph shows, that in most cases, subfeatures that are analyzed separately do not achieve better results than when they would have been analyzed combined with the remaining features of their corresponding feature category. Compared to analysis type I, the last four columns on the left side show that voting (no matter which version) achieves only slightly better results than the analysis of all features with a single classifier. As in previous charts, #A denotes the total number of authors within a dataset and #P denotes the total number of posts.

Single classifier. In “Voting all Features Separately without Emojis etc.” only those subfeatures with the best results are voted. Therefore this analysis does not include the results of the emoji, sentiment, topic modeling, grammar and typo analyses. As the weights for these features were already quite small within enquote Voting all Features Separately it is not surprising that the voting results of an analysis without them, do not change significantly. The results that are called “Voting Combined Feature Categories” are those that were discussed in the previous paragraph. Thus in this case, only four results of the four feature categories are voted. This version of voting seems to work best for large datasets as shown in Fig. D.32 and Fig. D.33. However, when the datasets are small, there is either no difference to the other two voting methods or the results are a little worse. More detailed results for each author achieved by the voting analysis can be found in Fig. D.24 to Fig. D.31.
Chapter 7

Conclusions

In the previous chapters, different theoretical basics about Authorship Attribution, Machine Learning, NLP and the Dark Web were discussed as well as their application to Dark Web forum posts. The following two chapters will now serve to summarize the most important core statements of this work and to give an insight into which research aspects and approaches in the context of this work can be investigated in the future.

7.1 Contributions

The focus of this thesis was on authorship attribution within multiple Dark Web forums. That AA is feasible within one forum was already shown by e.g., M. Spitters et al. in [69]. Thus the main research question of this thesis was whether it would be also possible beyond the boundaries of a single forum. AA within the Dark Web can become a curse for such users that try to protect their anonymity, and at the same time, can become a blessing for law enforcement groups or others that try to track users. Therefore, another motivation was to figure out to what extent AA threatens the anonymity of Dark Web users.

To achieve this goal, a crawler/scraper was developed that can extract posts from
different Dark Web forums. This crawler sends requests to a target service with such caution that the security mechanisms against automated website accesses of the service are hardly able to detect it. As a result, an extensive dataset of the four Dark Web forums DNM Avengers, The Hub, The Majestic Garden, and Dread could be created. By comparing public PGP-keys users were linked between these four forums so that six different datasets could be created, each containing the posts of linked users from two of the previous four mentioned forums.

Based on these six datasets, various features have been extracted that fall into the following four categories: lexical-based features, time-based features, stylometric-based features, and language model-based features. These four feature categories were analyzed with 18 different classifiers from scikit-learn to find the most appropriate one. On top of that, there are two main types of analyses within this thesis. Within the first, all classifiers are fed with a mixture of posts from both forums, whereas in the other case, the classifiers are trained with posts written in only one of the two forums and tested with the post from the remaining one. The reason for that was to be able to focus on the feasibility of AA based on texts written within different sources (first type) and on the other hand to be able to focus more on the features themselves and to find their weaknesses and strengths for the given task (second type). However, in each type, each forum combination is analyzed in different ways by either balancing the dataset or reducing the number of candidate authors to only those that wrote more posts than a specific threshold.

The results show that the likelihood that a post can be attributed correctly to its author decrease when the number of candidate authors increases. Furthermore, it seems that it is (in general) a good idea to reduce the number of posts from authors that wrote significantly more posts than the average. By this, it is more likely to better classify authors with only a few posts. However, even under the challenging conditions in this thesis that the analyzed texts come from multiple sources, there
is still a probability of at least 94% that the correct author of a post is one of the three most likely authors. This shows that AA is indeed a danger to the anonymity of Dark Web users across the boundaries of different forums. Therefore, all users that want to avoid getting linked via AA should keep the following aspects in mind: a post can most likely be assigned to its author if they tend to have an abnormal or very typical daily rhythm that is reflected in their online behavior, if they tend to write many long or short posts with the same structure, and also even when they write only a few but comparatively long posts with a large number of similar words.

7.2 Future Work

Several aspects could not be realized within this thesis and should be optimized or extended in future work. These aspects are mentioned in more detail below.

**Dataset:** the basis for all ML-based analyses is a data set. In this thesis, this data set is created by crawling four Dark Web forums. Unfortunately, only a few authors could be linked at all between these forums and those that could be linked rarely wrote more than 50 posts in both forums. Thus, it would be desirable to find more popular Dark Web forums that can be used as an additional data source for further validating the research presented. Currently, it is quite difficult to find such forums because a rising number of forums are going offline. However, there might be a time where this tendency is reversed and a larger dataset can be crawled.

**Features:** the quality of features is directly linked to the overall performance of the classifier. In this thesis, multiple different features were used, but unfortunately, not all of them could be optimized for the underlying problem. This was mainly due to the huge computation time that was needed for developing the LDA, Word2Vec, GloVe or FastText models. Therefore, there might be great potential when focusing
on the optimization of these language models to improve the overall results of AA. Also, the features containing information about grammar mistakes or typos could be improved by either developing a grammar checker or by experimenting with another tool. On top of that, it would be interesting to use similarity features like the Cosine similarity, Euclidean distance or Manhattan distance as a feature.

**Analysis:** the analysis in this thesis was quite extensive, which lead to the problem that there was not enough time to optimize all classifiers for all features. In some cases that might not be a problem (e.g., there is no need to optimize Naive Bayes-based classifiers for the time-based features because these features are categorized best by tree-based classifiers). However, especially in the case of the lexical-based features, language model-based features or when all features are combined would be interesting to know if one of the classifiers that needed too long to optimize (and therefore was canceled after 15 minutes), would achieve better results than those that completed the optimization process in time. Apart from this aspect, the analysis of ensembling methods like stacking, boosting, voting or bagging would also bear fruitful results. The last aspect that would be interesting are authors with similar usernames. In this thesis, PGP-keys were used as ground truth to link users within different forums. However, other research papers focus only on usernames to link authors (e.g., Ho and Ng in [29]). The problem is, that usernames can be chosen without any restriction which means that a user with the username rabit123 from forum A does not have to be the user with the username rabit123 within forum B. Therefore, linking users by their username might not be as accurate as to link them via PGP-keys but it would be interesting to figure out for how many of them this way of linking would actually work.
Bibliography


Appendix A

Artificial Intelligence

A.1 Latent Dirichlet Allocation

The input of the algorithm is a word-frequency based matrix (see crefbow) and thus all words of the text corpus that remained after a preprocessing step (see the left side of Fig. 3.4). The output is a word × topic matrix. For each of the $K$ topics (where $K$ is a predefined value by the user), the probability for each word is calculated, with that it can be assigned to the topic. The basic idea of the algorithm itself is that documents contain a random mixture over different latent topics and each topic is made out of a distribution over words [17].

Therefore the algorithm iterates over each word $w_{jt}, t \in 1, ..., N_j$ within each document $j, j \in 1, ..., M$ (boxes of the plate notation in Fig. 3.4 representing repetitions). \(\alpha\) and \(\beta\) are parameters of the Dirichlet distribution (a distribution of distributions) where \(\alpha\) is needed to control the per-document topic distribution and \(\beta\) to control the per-topic word distribution. Or in other words, a high value for \(\alpha\) indicates that it is likely that each document contains a mixture of most of the $K$ topics whereas a low value represents the fact that it is more likely that each document contains only a small fraction of the $K$ topics. The same applies to the parameter \(\beta\). Here, a high
value expresses a high probability, that each topic consists out of a mixture of most of the words provided in the text corpus whereas a low value implies that each topic is represented by only a few words. Thus a high $\alpha$ will make all documents appear more similar to each other whereas a high $\beta$ will result in that all topics are more similar to each other and vice versa.

### A.2 Soft-Margin SVMs

Unfortunately, linear separability like in the section before is not always given. Sometimes data is linearly inseparable like in the example shown in Fig. 3.6. There are two “outliers”, each surrounded by an extra circle, that exceed the minimum distance to the hyperplane. There are two cases of outliers: those that are placed before the hyperplane and those that are placed behind it.

In the first case, it is $0 < \xi_i < 1$ for distance $\xi$. The outlier can still be classified correctly, although it is closer than $\frac{1}{\|w\|}$ to the hyperplane. In contrast, the distance $\xi_j$ of the red outlier to its class is too large ($\xi_j \geq 1$) to be classified correctly. In this case, the SVM will return an incorrect result. This ends up in the problem to keep the sum of all $\xi_i$ as small as possible, but to find a hyperplane that is as far away from the two classes as possible. As before, this can be defined as an optimization problem [11].

\[
\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i^p \\
\text{subject to } y_i((w \cdot x_i + b) \geq 1 - \xi_i \land \xi_i \geq 0, i = 1, ..., n}
\]

The parameter $C \geq 0$ stands for the trade-off between the largest distance or smallest $\|w\|^2$ and the minimization of the slack penalty $\sum_{i=1}^{n} \xi_i^p$, $\xi = (\xi_1, ..., \xi_n)^\top$, $p \geq 1$. The choice of the values for $p$ determine whether outliers are suppressed more or less aggressively [49].

The normal vector $w$ can be represented as in the previous case from a linear combi-
nation of the training data \((x_1, \ldots, x_n)\). If \(\alpha \neq 0\) applies, then \(x_i\) is a support vector. However, there are two different types of support vectors at this point. In the first case \(y_i(w \cdot x_i + b) = 1 - \xi_i\) applies. If \(\xi_i = 0\), then \(y_i(w \cdot x_i + b) = 1\) applies. In this case \(x_i\) is located exactly in the hyperplane. However, if \(\xi_i \neq 0\) applies, then \(x_i\) is an outlier.

The decision function received after solving the optimization problem is the same as in Eq. (3.18). Therefore, also the calculation for \(b\) is the same as before, but with the additional condition that the support vector \(x_i\), which should be used to calculate \(b\), must be somewhere between \(\alpha_i\) (the Lagrange multipliers) and \(C\) (the tradeoff parameter) \([49]\).

### A.3 Transformation of Continuous Values within Decision Trees

A major disadvantage of the approach mentioned above is that it only works with discrete values. Fortunately, a small modification is sufficient to be also able to use features with non-discrete values. The basic idea is to convert all values \(x\) of all features into boolean values using a threshold value of \(\theta\). This \(\theta\) should be chosen in a way that for all \(x\) that belong to \(x < \theta\) become \textit{true} whereas all others become \textit{false}. However, the question is how to choose \(\theta\) best. A very reasonable solution would be to sort all values for \(x\) of the certain feature by size and then calculate \(n-1\) thresholds according to the formula \(\frac{(x_i + x_{i+1})}{2}\). From these \(n-1\) \(\theta_i\)’s, the information content, which the feature would have after converting its values, can be calculated. The threshold value \(\theta_i\) that gives the feature the highest information content will be used for the conversion \([39]\).

The problem with this method is that it is extremely calculation-intensive and that many of the calculated thresholds are not needed at all. The following procedure
Fig. A.1. Threshold ($\theta_i$) computation method to convert continues values into boolean values by sorting all values of a feature and also take their corresponding class into account \[39\].

offers a solution: As before, all existing values of a feature are sorted first. On top of that, the algorithm labels each value with its corresponding class. The solution to the problem is that a threshold value never lies between the values of the same class, but always on the boundary between two different classes as shown in Fig. A.1. This rule reduces the possible thresholds considerably \[39\].

### A.4 Naive Bayes and Continuous Attributes

In the previous sections, the focus was on attributes with discrete values. Unfortunately, many applications in the real world cannot be expressed by these values. Hence, continuous values are needed, but they have an infinite number of possible values. Thus, when computing the probability of a specific value of a continuous attribute, the result will become extremely small \[39\].

To overcome this problem, one solution is to discretize the continuous attributes. This can be done by splitting the range of the values up into several slots/intervals $b_i$ and then compute the probabilities for each $b_i$ that a value $x$ belongs to it (see Fig. A.2). However, this might be an easy solution to the problem, but unfortunately, lots of information can get lost by this technique \[39\]. Therefore, another way to deal with continuous attributes is to reduce the size of the intervals until the steps of the “step-function” like in Fig. A.2 becomes infinitesimally small as shown in Fig. A.3. Hereby, the step-function turns into a continuous function $p(x)$. A high value of $p(x)$ indicates that many examples are close to $x$ while a low value of $p(x)$
indicates the opposite. This is why the name of $p()$ is *Probability Density Function* (PDF). In the following $p(x)$ is the pdf at point $x$ whereas $P(x)$ is the probability of the discrete value $x$ and $p_{y_j}$ is the pdf of class $y_j$. Fortunately, it is possible to adapt the Bayes formula to continuous attributes by using the pdf. The only thing that needs to be changed is to replace $P(x|y_j)$ by $p_{y_j}$ and $P(x)$ by $p(x)$ respectively. Thus, the “new” Bayesian formula for a dataset with more than one attribute is:

$$p_{y_j}(x) = \prod_{i=1}^{n} p_{y_j}(x_i) \quad (A.3)$$
Appendix B

Dataset

Fig. B.1. Captchas that need to be solved before being able to access the Dark Web forum Dread. The left figure shows the captcha without any user input whereas the right figure shows the correctly solved captcha.

Fig. B.2. Unsolved captcha from Dread.

Fig. B.3. Solved captcha from Dread.
## Appendix C

### Features

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical</td>
<td>Count Vectoriser</td>
</tr>
<tr>
<td>Lexical</td>
<td>TF-IDF</td>
</tr>
<tr>
<td>Language Model</td>
<td>Word2Vector</td>
</tr>
<tr>
<td>Language Model</td>
<td>GloVe</td>
</tr>
<tr>
<td>Language Model</td>
<td>FastText</td>
</tr>
<tr>
<td>Language Model</td>
<td>Positive sentiment</td>
</tr>
<tr>
<td>Language Model</td>
<td>Negative sentiment</td>
</tr>
<tr>
<td>Language Model</td>
<td>Neutral sentiment</td>
</tr>
<tr>
<td>Language Model</td>
<td>Compound (pos., neg., neu. sentiment)</td>
</tr>
<tr>
<td>Language Model</td>
<td>LDA</td>
</tr>
<tr>
<td>Language Model</td>
<td>NMF</td>
</tr>
<tr>
<td>Stylometric</td>
<td>11 Emoji categories</td>
</tr>
<tr>
<td>Stylometric</td>
<td>306 grammar/spelling error-groups</td>
</tr>
<tr>
<td>Stylometric</td>
<td>701 possible typos (a/b,b/a,a/c,c/a, etc.)</td>
</tr>
<tr>
<td>Stylometric</td>
<td>35 POS tags (see Table C.2)</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of characters per word</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of capital letters</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of small letters</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of punctuation marks</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of abbreviations</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median, stdev. of the number of words with both cases</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of lowercase-words</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of uppercase words</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of numbers</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of words</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of spaces</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median, stdev. # of sentences starting with a capital letter</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the Lexical richness</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Sum, avg., median and stdev. of the number of different POS tags</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Number of Sentences</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Number of Lines</td>
</tr>
<tr>
<td>Stylometric</td>
<td>Number of invisible Characters</td>
</tr>
<tr>
<td>time</td>
<td>Minute</td>
</tr>
<tr>
<td>time</td>
<td>Hour</td>
</tr>
<tr>
<td>time</td>
<td>Day</td>
</tr>
<tr>
<td>time</td>
<td>Month</td>
</tr>
<tr>
<td>time</td>
<td>Year</td>
</tr>
<tr>
<td>time</td>
<td>Weekday</td>
</tr>
</tbody>
</table>

Table C.1: List of all features used within this analysis.
Table C.2: List of all POS tags considered within the stylometric features.

<table>
<thead>
<tr>
<th>POS Tag</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>coordinating conjunction</td>
</tr>
<tr>
<td>CD</td>
<td>cardinal digit</td>
</tr>
<tr>
<td>DT</td>
<td>determiner</td>
</tr>
<tr>
<td>EX</td>
<td>existential there</td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
</tr>
<tr>
<td>IN</td>
<td>preposition/subordinating conjunction</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective</td>
</tr>
<tr>
<td>JJR</td>
<td>adjective, comparative</td>
</tr>
<tr>
<td>JJS</td>
<td>adjective, superlative</td>
</tr>
<tr>
<td>LS</td>
<td>list marker</td>
</tr>
<tr>
<td>MD</td>
<td>modal</td>
</tr>
<tr>
<td>NN</td>
<td>noun, singular</td>
</tr>
<tr>
<td>NNS</td>
<td>noun plural</td>
</tr>
<tr>
<td>NNP</td>
<td>proper noun, singular</td>
</tr>
<tr>
<td>NNPS</td>
<td>proper noun, plural</td>
</tr>
<tr>
<td>PDT</td>
<td>predeterminer</td>
</tr>
<tr>
<td>POS</td>
<td>possessive ending</td>
</tr>
<tr>
<td>PRP</td>
<td>personal pronoun</td>
</tr>
<tr>
<td>PRP$</td>
<td>possessive pronoun</td>
</tr>
<tr>
<td>RB</td>
<td>adverb</td>
</tr>
<tr>
<td>RBR</td>
<td>adverb, comparative</td>
</tr>
<tr>
<td>RBS</td>
<td>adverb, superlative</td>
</tr>
<tr>
<td>RP</td>
<td>particle</td>
</tr>
<tr>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>UH</td>
<td>interjection</td>
</tr>
<tr>
<td>VB</td>
<td>verb, base form</td>
</tr>
<tr>
<td>VBD</td>
<td>verb, past tense</td>
</tr>
<tr>
<td>VBG</td>
<td>verb, gerund/present participle</td>
</tr>
<tr>
<td>VBN</td>
<td>verb, past participle</td>
</tr>
<tr>
<td>VBP</td>
<td>verb, sing. present</td>
</tr>
<tr>
<td>VBZ</td>
<td>verb, 3rd person sing. present</td>
</tr>
<tr>
<td>WDT</td>
<td>determiner that starts with wh</td>
</tr>
<tr>
<td>WP</td>
<td>pronoun that starts with wh</td>
</tr>
<tr>
<td>WPS$</td>
<td>possessive pronoun that starts with wh</td>
</tr>
<tr>
<td>WRB</td>
<td>adverb that starts with wh</td>
</tr>
</tbody>
</table>
Appendix D

Authorship Attribution Analysis

D.1 Hardware

For each analysis one cluster node with the following specifications is used. The cluster is operated by the Platform for Scientific Computing at Bonn-Rhein-Sieg University [26].

- Intel Xeon Gold 6130 and Intel Xeon Gold 6130-F at 2.1 GHz, each with in total 64 hardware threads
- 192 GB DDR4-2466 memory
- 480 GB SSD
- 100 Gb/s Intel Omni-Path through a Xeon Gold 6130-F
- L1 data cache: 32 KB, 8-way set associative, write-back, 64 bytes/line
- L2 unified cache: 1 MB, 8-way set associative, write-back, 64 bytes/line, exclusive cache
- L3 cache: 22 MB, 16-way associative, write-back, exclusive cache
- 6 memory channels per processor, theor. memory bandwidth 127.8 GB/s per processor
- peak performance is 54.4 GigaFlops / processor core thread

D.2 Tables and Images
Table D.1: Listing of all 24 analyses of type I

<table>
<thead>
<tr>
<th>Analysis type</th>
<th>Forum combination</th>
<th>Number of posts</th>
<th>Balanced or not</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-1</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-2</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-3</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-4</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>500+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>1000+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-5</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>500+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>1000+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-6</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>500+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>1000+</td>
<td></td>
<td>balanced</td>
</tr>
</tbody>
</table>
Table D.2: Listing of all 28 analyses of type II

<table>
<thead>
<tr>
<th>Analysis type</th>
<th>Forum combination</th>
<th>Number of posts</th>
<th>Balanced or not</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-1</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-2</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>200+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-3</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>200+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-4</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>200+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>400+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-5</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>200+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>400+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td>FC-6</td>
<td>all</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>200+</td>
<td></td>
<td>balanced</td>
</tr>
<tr>
<td></td>
<td>400+</td>
<td></td>
<td>balanced</td>
</tr>
</tbody>
</table>
Fig. D.1. Overview of the F1-Score achieved within the analyses of posts from all authors with 500 or more posts of the remaining three forum combinations (FC-4 to FC-6) for each feature category with analysis type I by using either an unbalanced dataset (left) or balanced dataset (right). # A denotes the number of authors within the corresponding forum combination and # P the number of posts. There is either none or just one author left within the other three forum combinations when rising the threshold of posts to this level and thus they cannot be used for an analysis.

Fig. D.2. Overview of the F1-Score achieved within the analyses of posts from all authors with 1000 or more posts of the remaining three forum combinations (FC-4 to FC-6) for each feature category with analysis type I by using either an unbalanced dataset (left) or balanced dataset (right). # A denotes the number of authors within the corresponding forum combination and # P the number of posts. There is either none or just one author left within the other three forum combinations when rising the threshold of posts to this level and thus they cannot be used for an analysis.
Fig. D.3. Overview of the F1-Score achieved within the analyses of posts from all authors with 200 or more posts of the remaining five forum combinations (FC-2 to FC-6) for each feature category with analysis type II by using either an unbalanced dataset (left) or balanced dataset (right). # A denotes the number of authors within the corresponding forum combination and # P the number of posts. There is no author left within the remaining forum combination when rising the threshold of posts to this level and thus it cannot be used for an analysis.

Fig. D.4. Overview of the F1-Score achieved within the analyses of posts from all authors with 400 or more posts of the remaining five forum combinations (FC-4 to FC-6) for each feature category with analysis type II by using either an unbalanced dataset (left) or balanced dataset (right). # A denote the number of authors within the corresponding forum combination and # P the number of posts. There is either none or just one author left within the remaining three forum combination when rising the threshold of posts to this level and thus they cannot be used for an analysis.
Table D.3: Absolute frequencies with which the listed classifiers achieved the best result for the respective feature category in the 52 analyses of this thesis.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Time</th>
<th>Stylometric</th>
<th>Lexical</th>
<th>Language Models</th>
<th>All Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>BernoulliNB</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ComplementNB</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DecisionTreeClassifier</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtraTreesClassifier</td>
<td>22</td>
<td>11</td>
<td>0</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>GaussianNB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>GaussianProcessClassifier</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>KNeighborsClassifier</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LinearSVC</td>
<td>0</td>
<td>26</td>
<td>7</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>MLPClassifier</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>MultinomialNB</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>NuSVC</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>PassiveAggressiveClassifier</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>Perceptron</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>RandomForestClassifier</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SGDClassifier</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>SVC</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. D.5. The detailed results of all authors within the balanced and unbalanced analysis (type II) of FC-5 (left) and FC-6 (right) by using **time-based features** only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).
Fig. D.6. The detailed results of all authors within the balanced and unbalanced analysis (type II) of FC-5 (left) and FC-6 (right) by using **stylometric-based features** only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).

Fig. D.7. The detailed results of all authors within the balanced and unbalanced analysis (type II) of FC-5 (left) and FC-6 (right) by using **lexical features** only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).
Fig. D.8. The detailed results of all authors within the balanced and unbalanced analysis (type II) of FC-5 (left) and FC-6 (right) by using language model-based features only. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).

Fig. D.9. The detailed results of all authors within the balanced and unbalanced analysis (type II) of FC-5 (left) and FC-6 (right) by using all features combined. The diagram shows the impact of the number of posts as well as the sum of words an author has written within the dataset on this feature category. The authors are sorted by the number of posts from the left (few posts) to the right (many posts).
Fig. D.10. Statistics on the post lengths of each author from FC-5 and within the various subanalyses (see titles of the individual sub-graphs) of analysis type I. The authors are sorted in ascending order based on their numeration.
Fig. D.11. Statistics on the post lengths of each author from **FC-6** and within the various subanalyses (see titles of the individual sub-graphs) of analysis **type I**. The authors are sorted in ascending order based on their numeration.
Fig. D.12. Statistics on the post lengths of each author from FC-5 in the training set and test set within the three unbalanced subanalyses (see titles of the individual sub-graphs) in analysis type II. The authors are sorted in ascending order based on their numeration.
Fig. D.13. Statistics on the post lengths of each author from FC-5 in the training set and test set within the three balanced subanalyses (see titles of the individual sub-graphs) in analysis type II. The authors are sorted in ascending order based on their numeration.
Fig. D.14. Statistics on the post lengths of each author from FC-6 in the training set and test set within the three unbalanced subanalyses (see titles of the individual sub-graphs) in analysis type II. The authors are sorted in ascending order based on their numeration.
Fig. D.15. Statistics on the post lengths of each author from FC-6 in the training set and test set within the three balanced subanalyses (see titles of the individual sub-graphs) in analysis type II. The authors are sorted in ascending order based on their numeration.
Fig. D.16. Results for each author achieved with all features combined within analysis type of FC-5, with an unbalanced data set and when considering all authors.
Fig D.17. Results for each author achieved with all features combined within analysis type of FC-5 with a balanced data set (only the longest posts are chosen) and when considering all authors.
Fig. D.18. Results for each author achieved with all features combined within analysis type of FC-6 with an unbalanced data set and when considering all authors.
Fig. D.19. Results for each author achieved with all features combined within analysis type of FC-6 with a balanced data set (only the longest posts are chosen) and when considering all authors.
Fig. D.20. Results for each author achieved with all features combined within analysis type II of FC-5 with a balanced data set (only the longest posts are chosen) and when considering all authors.
Fig. D.21. Results for each author achieved with all features combined within analysis type II of FC-5 with a balanced data set (only the longest posts are chosen) and when considering all authors.
Fig. D.22. Results for each author achieved with all features combined within analysis type II of FC-6 with an unbalanced data set and when considering all authors.
Fig. D.23. Results for each author achieved with all features combined within analysis type II of FC-6 with a balanced data set (only the longest posts are chosen) and when considering all authors.
Fig. D.24. Results for each author achieved when voting the results of all feature categories within analysis type I of FC-5 with an unbalanced data set and when considering all authors.
Fig. D.25. Results for each author achieved when voting the results of all feature categories within analysis type I of FC-5 with a balanced data set and when considering all authors.

Confusion matrix

Scores and ranking

<table>
<thead>
<tr>
<th>Author</th>
<th>Posts for training</th>
<th>Posts for testing</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Top1</th>
<th>Top2</th>
<th>Top3</th>
<th>Top4</th>
<th>Top5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>73</td>
<td>31</td>
<td>0.92</td>
<td>0.71</td>
<td>0.84</td>
<td>71%</td>
<td>16%</td>
<td>6%</td>
<td>2%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A1</td>
<td>73</td>
<td>31</td>
<td>0.88</td>
<td>0.70</td>
<td>0.77</td>
<td>77%</td>
<td>10%</td>
<td>6%</td>
<td>2%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A2</td>
<td>73</td>
<td>31</td>
<td>0.91</td>
<td>0.84</td>
<td>0.89</td>
<td>94%</td>
<td>3%</td>
<td>0%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A3</td>
<td>72</td>
<td>32</td>
<td>0.71</td>
<td>0.75</td>
<td>0.73</td>
<td>75%</td>
<td>6%</td>
<td>3%</td>
<td>6%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A4</td>
<td>73</td>
<td>31</td>
<td>0.75</td>
<td>0.84</td>
<td>0.81</td>
<td>84%</td>
<td>4%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A5</td>
<td>72</td>
<td>32</td>
<td>0.86</td>
<td>0.84</td>
<td>0.85</td>
<td>94%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A6</td>
<td>73</td>
<td>31</td>
<td>0.83</td>
<td>0.91</td>
<td>0.87</td>
<td>87%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A7</td>
<td>73</td>
<td>31</td>
<td>0.83</td>
<td>0.91</td>
<td>0.87</td>
<td>87%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A8</td>
<td>73</td>
<td>31</td>
<td>0.79</td>
<td>0.67</td>
<td>0.72</td>
<td>72%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A9</td>
<td>73</td>
<td>31</td>
<td>0.76</td>
<td>0.61</td>
<td>0.69</td>
<td>69%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>A10</td>
<td>73</td>
<td>31</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
<td>84%</td>
<td>3%</td>
<td>0%</td>
<td>0%</td>
<td>3%</td>
<td>1%</td>
</tr>
<tr>
<td>A11</td>
<td>72</td>
<td>32</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>91%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A12</td>
<td>73</td>
<td>31</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
<td>78%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A13</td>
<td>73</td>
<td>31</td>
<td>0.81</td>
<td>0.81</td>
<td>0.81</td>
<td>81%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A14</td>
<td>72</td>
<td>32</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
<td>84%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A15</td>
<td>73</td>
<td>31</td>
<td>0.81</td>
<td>0.81</td>
<td>0.81</td>
<td>81%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>A16</td>
<td>73</td>
<td>31</td>
<td>0.83</td>
<td>0.87</td>
<td>0.86</td>
<td>86%</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>3%</td>
<td>3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>weighted</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>85%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>Median</td>
<td>0.84</td>
<td>0.84</td>
<td>0.83</td>
<td>83%</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>Stdv</td>
<td>0.07</td>
<td>0.07</td>
<td>0.08</td>
<td>8%</td>
<td>3%</td>
<td>2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Sum</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
|      | 1237               | 531  | 85%       | 3%      | 2%       | 1%   | 4%
Fig. D.26. Results for each author achieved when voting the results of all feature
categories within analysis type I of FC-6 with an unbalanced data set and when
considering all authors.

<table>
<thead>
<tr>
<th>Predicted label</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>A9</th>
<th>A10</th>
<th>A11</th>
<th>A12</th>
<th>A13</th>
<th>A14</th>
<th>A15</th>
<th>A16</th>
<th>A17</th>
<th>A18</th>
<th>A19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>A2</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>A3</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>A4</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td>A5</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>A6</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
</tr>
<tr>
<td>A7</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
</tr>
<tr>
<td>A8</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
</tr>
<tr>
<td>A9</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
</tr>
<tr>
<td>A10</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
<td>128</td>
<td>129</td>
</tr>
<tr>
<td>A11</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
</tr>
<tr>
<td>A12</td>
<td>131</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
</tr>
<tr>
<td>A13</td>
<td>141</td>
<td>142</td>
<td>143</td>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
</tr>
<tr>
<td>A14</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
</tr>
<tr>
<td>A15</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
</tr>
<tr>
<td>A16</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
</tr>
<tr>
<td>A17</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
</tr>
<tr>
<td>A18</td>
<td>191</td>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
<td>208</td>
<td>209</td>
</tr>
<tr>
<td>A19</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
</tr>
</tbody>
</table>

Scores and Ranking:

- High scores indicate a strong tendency towards the predicted label.
- Lower scores suggest a weaker tendency or uncertainty.

Analysis Type: FC-6
Unbalanced Data Set
Fig. D.27. Results for each author achieved when voting the results of all feature categories within analysis type I of FC-6 with a balanced data set and when considering all authors.
Fig. D.28. Results for each author achieved when voting the results of all feature categories within analysis type II of FC-5 with an unbalanced data set and when considering all authors.

149
Fig. D.29. Results for each author achieved when voting the results of all feature categories within analysis type II of FC-5 with a balanced data set and when considering all authors.
Fig. D.30. Results for each author achieved when voting the results of all feature
categories within analysis type II of FC-6 with an unbalanced data set and when
considering all authors.
Fig. D.31. Results for each author achieved when voting the results of all feature categories within analysis type II of FC-6 with a balanced data set and when considering all authors.
Fig. D.32. Detailed overview of all results (incl. subfeatures) achieved by analysis type I (unbalanced) when analyzing FC-5 and FC-6. The graph shows that subfeatures that are analyzed separately do not achieve better results than when they would have been analyzed combined with the remaining features of their corresponding feature category. On top of that, the last four columns on the left side show that voting (no matter which version) achieves better results than the analysis of all features with a single classifier. As in previous charts, #A denotes the total number of authors within a dataset and #P denotes the total number of posts.

Fig. D.33. Detailed overview of all results (incl. subfeatures) achieved by analysis type II (unbalanced) when analyzing FC-5 and FC-6. The graph shows that in most cases, subfeatures that are analyzed separately do not achieve better results than when they would have been analyzed combined with the remaining features of their corresponding feature category. Compared to analysis type I, the last four columns on the left side show that voting (no matter which version) achieves only slightly better results than the analysis of all features with a single classifier. As in previous charts, #A denotes the total number of authors within a dataset and #P denotes the total number of posts.
Vita

Candidate’s full name:
Britta Sennewald

University attended (with dates and degrees obtained):
Muthesius University of Fine Arts and Design, Kiel, Germany.
Bachelor of Arts (Communication Design), 02-2014;
Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany.
Bachelor of Science (Computer Science), 08-2017.

Publications:
Conference Presentations: