5-2024

Multimodal stylometry: A novel approach for authorship identification.

Glory O. Adebayo
University of Louisville

MULTIMODAL STYLOMETRY: A NOVEL APPROACH FOR AUTHORSHIP IDENTIFICATION

By

Glory Olajide Adebayo
B.Sc. Covenant University, Ota, Nigeria (2012)
M.Sc. Robert Gordon University, Aberdeen, Scotland (2015)

A Dissertation submitted to the J.B. Speed School of Engineering, University of Louisville, in Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy
in Computer Science and Engineering

Department of Computer Science and Engineering,
University of Louisville,
Louisville, Kentucky

May 2024

MULTIMODAL STYLOMETRY: A NOVEL APPROACH FOR AUTHORSHIP IDENTIFICATION

By

Glory Olajide Adebayo
B.Sc. Covenant University, Ota, Nigeria (2012)
M.Sc. Robert Gordon University, Aberdeen, Scotland (2015)

A Dissertation approved on

April 18, 2024

By the following Dissertation Committee

_____
Dr Roman V. Yampolskiy
Committee Chair

_____
Dr. Olfa Nasraoui
Committee Member

_____
Dr. Adrian Lauf
Committee Member

_____
Dr. Michael Losavio
Committee Member

DEDICATION

This dissertation is dedicated with profound gratitude and heartfelt appreciation to my pillar of strength, my wife, Bisola Asaolu. Her unwavering support, understanding, and encouragement have been my constant companions throughout the challenging journey of my doctorate program.

I extend my deepest appreciation to my parents, Rt Revd. Jide Adebayo and Mrs. Foluke Adebayo, whose love, wisdom, and sacrifices have laid the foundation for my academic pursuits. Their guidance and encouragement have been instrumental in shaping my scholarly path.

To my siblings, Ibukun Adebayo, Dr. Ololade Folayan (Dr. T), and Oreoluwa Adebayo, I express my gratitude for standing by me with unyielding support. Your encouragement and belief in my abilities have been a source of inspiration, making this academic journey more meaningful.

This dissertation is not just a testament to my academic achievements but also a reflection of the collective support and love bestowed upon me by my cherished family. I am profoundly grateful for the bond we share, and the strength derived from our unity.

In dedicating this work to my loved ones, I acknowledge the sacrifices and contributions each of you has made, and I am genuinely thankful for the unwavering belief you have in me. This achievement is as much yours as it is mine.

ACKNOWLEDGEMENT

I extend my deepest gratitude to the many individuals who have played an integral role in the completion of this doctoral dissertation, marking the culmination of an enriching and challenging journey. Their unwavering support, guidance, and encouragement have been instrumental in shaping this research endeavor.

First and foremost, I express my heartfelt appreciation to my advisor, Dr. Roman Yampolskiy for his exceptional mentorship and scholarly insight throughout every stage of this research. His dedication, expertise, and constructive feedback have been invaluable in refining the research questions and methodologies, fostering both intellectual and personal growth.

I am indebted to the members of my dissertation committee, Doctors Nasraoui, Lauf and Losavio, for their valuable contributions, critical feedback, and insightful suggestions that have significantly enhanced the quality and rigor of this dissertation. Their diverse expertise has provided a comprehensive perspective that has been instrumental in shaping the research.

I extend my sincere thanks to the Department of Computer Science and Engineering at University of Louisville for providing a conducive academic environment, resources, and opportunities for intellectual exploration. The support from faculty members, administrative staff, and fellow researchers has created a nurturing community that has enriched my academic experience.

My gratitude extends to my family for their unwavering support, understanding, and encouragement throughout this demanding journey. Their belief in my abilities and resilience in times of challenges have been a source of strength and motivation.

I am grateful to my friends and colleagues who have shared their insights, provided constructive critiques, and offered words of encouragement. Their camaraderie has made this academic pursuit not only intellectually rewarding but also personally fulfilling.

Lastly, I acknowledge the support of my supervisors and colleagues at River Road Asset Management for their support as I wound down on completing this journey.

This dissertation is a collective effort, and I am profoundly grateful to everyone who has contributed to its realization. Their support has been an indispensable part of this scholarly journey, and I am truly thankful for their unwavering belief in my abilities.

ABSTRACT

MULTIMODAL STYLOMETRY: A NOVEL APPROACH FOR AUTHORSHIP IDENTIFICATION

Glory O. Adebayo

April 18, 2024

This dissertation introduces multimodal stylometry, a novel approach to authorship identification that integrates text and source code features for a comprehensive understanding of an author's unique style. Traditional stylometric methods have primarily focused on either text stylometry or source code stylometry, thereby neglecting the potential insights that multimodality may provide. This research aims to bridge this gap by proposing a framework that combines textual and source code data to enhance the accuracy and reliability of authorship identification.

The study begins by reviewing existing literature on authorship identification and stylometry, highlighting the limitations of unimodal approaches. Leveraging recent advancements in multimodal biometrics and feature fusion, the research introduces a methodology that extracts stylometric features from written text and source code. These multimodal features are then integrated using an extended feature fusion technique that introduces an extra layer of feature selection.

To validate the proposed approach, a diverse dataset comprising texts and corresponding source code data from various authors is curated. The dissertation explores the effectiveness of multimodality when compared to unimodality.

Furthermore, the research investigates the transferability of the proposed multimodal stylometry framework in distinguishing AI and Human generated text and source code.

The findings not only advance authorship identification techniques but also hold implications for applications in forensic linguistics, digital humanities, and content analysis. Ultimately, this research underscores the significance of multimodal stylometry in estimating the identity of an author

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Stylometry is a growing field of study that analyses, and quantifies the stylistic features exhibited by individuals, be they authors, artists, singers, or even programmers. The primary goal of this discipline is to discern and attribute the authorship of a work, pinpoint its origin, or unravel the distinct attributes of the creator. Over recent times, stylometry has garnered a burgeoning level of attention and interest, primarily due to its various applications across diverse domains, including but not limited to forensic science, literary studies, and computational linguistics.

Stylometric analysis typically involves the extraction of stylistic or linguistic features from a body of work usually text. These features are then compared across different text to identify patterns and similarities that can be used to determine authorship, origin or estimate the attributes of the author (gender, native language [1], [2], [3].

1.2 The Human Stylome Hypothesis

Recently, a couple of researchers have put forward the “human stylome hypothesis” [4]. This intriguing hypothesis suggests that authors can be distinguished from one another through the quantification of specific features within their written works. However, the effectiveness of this approach is contingent on a case-by-case assessment, often closely tied to the context in which it is applied. In practical terms, this implies that the differentiation process is primarily relevant when comparing a known author or group of authors, characterized by established stylometric signatures, with an unknown text or corpus of text. This scenario is commonly referred to as a "close-world problem" since we possess knowledge of certain variables within the system. In this context, the approach serves to mitigate potential sources of noise in the dataset and offers a means to corroborate or challenge results obtained through other authorship identification methods, such as historical or documentary evidence.

However, some scholars have proposed an even more stringent criterion for author differentiation. They contend that authors should exhibit stylometric signatures that are not only identifiable but also invariant [5]. In other words, these signatures should remain consistent and unaltered over time and across various contexts. This concept poses a significant challenge, given that individuals often adapt and modify their writing style in response to different contexts and genres. For instance, an author's writing style may vary when composing a formal document as opposed to an informal communication. Consequently, the pursuit of such highly consistent signatures, both theoretically and practically, necessitates their applicability across

languages and linguistic boundaries. This ambitious aim seeks to uncover the core of an author's unique writing style, transcending temporal and contextual constraints.

As we investigate research that has been done in the field of stylometry, some classes of features have been identified for analysis [6]. These features are typically categorized into three main classes: lexical features, which measure the richness and diversity of an individual's vocabulary or domain; syntactic features, encompassing the structural aspects of a language, such as n-grams, parts of speech, punctuation usage, and non-context-sensitive function words; and semantic features, which delve into the meaning-based facets of language. Among these, the most highly prized are those that prove resistant to subconscious manipulation. Stylometric researchers often refer to this coveted entity as the "author's stylometric print," likening it to the writing equivalent of a handwritten signature or fingerprint. It is often discerned from latent data, which emerges from the author's unconscious and habitual linguistic behaviors, providing a valuable anchor for stylometric analysis to identify the unique and immutable style of an author. For example, research done by Montero et al [7] showed that the gender of an author could be identified based on how emotion is expressed in writing. The work showed that women tend to be more contextual, personal, and emotional than male writing. While male writing tends to be typically more impersonal, formal, and judgmental.

1.3 Closed world vs open world

In a closed-world scenario, the fundamental premise revolves around the binary classifications of True or False. It's a stark dichotomy where we are tasked with either successfully identifying the author, marking it as True, or encountering a situation where we cannot establish authorship, labeled as False. This outcome hinges on the overarching objective of the system, which essentially dictates whether we can pinpoint the author's identity. In a close world scenario, all the authors are known and are part of the training set used to build the model. Any writing from an author not in the training set would probably be misclassified as an author in the training data.

On the other hand, within an open-world scenario, the landscape of possibilities extends far beyond the confines of mere True or False classifications. Here, we grapple with the nuanced inclusion of an "unknown" element. This notion implies that authorship identification doesn't always fall into a straightforward binary paradigm. In this context, the elusive "unknown" introduces a compelling layer of complexity, suggesting that the true author might not necessarily belong to the set of initial suspects (training data). It challenges us to explore the realm of potential authorship beyond the confines of our preconceived classifications, introducing an intriguing element of unpredictability into the equation.

1.4 Multimodal stylometry

The world we inhabit is a rich combination of sensory experiences, involving diverse modalities. Our senses are engaged in multiple ways: we hear sounds, touch and feel the texture of objects, and see the vibrant world around us. Modality,

in this context, is the lens through which we engage with our surroundings and the medium through which experiences manifest. It refers to the specific way something is both experienced and communicated. Multimodality can therefore be referred to as the combination of different modalities (sight, sound, images, text, source code, music etc.) to produce a given output or message.

The intricacies of multimodality are instrumental in our daily lives and across various domains. From the multi-layered narratives of literature and film to the immersive experiences in virtual reality, the interplay of modalities is at the heart of effective communication. Moreover, in the digital age, the fusion of text, images, and sound on websites, social media, and other digital platforms has become the norm. The significance of multimodality extends to education, accessibility, and design, where the deliberate integration of multiple modalities can facilitate comprehension and engagement, enriching our interactions with the world and with one another. While the problems of identifying the authors of source code [8], [9] and written text [10], [11] has been tackled individually, less attention has been paid to multi-modality to the field of stylometry or combining the features of textual documents with features from source code documents to improve the classification accuracy of Authorship identification. Multimodality has been shown to improve accuracy in some other areas of research. (Biometrics [12], [13]).

Multimodal stylometry is therefore a novel approach in stylometry that involves the analysis of multiple modalities or sources of information. We propose that this approach has the potential to capture more comprehensive and nuanced stylistic

information than traditional unimodal stylometry. In the context of this research, it pertains to the concurrent analysis of text and source code data to identify authorship patterns. This approach offers an exciting prospect, one that holds the potential to leverage the advantages of stylistic characteristics of two modalities compared to unimodal stylometry. The combination of these modalities can improve authorship profiling, identification and plagiarism by considering a broader range of expressive elements.

Multimodality has already demonstrated its efficacy in diverse studies. For instance, Agrawal et al. [14] introduced the utilization of multimodal approaches, incorporating both audio and video data, to achieve successful personality recognition. This success serves as a testament to the vast potential of multimodal stylometry in authorship analysis.

One major challenge that might be encountered in multimodal stylometry is the integration of different modalities and features known as multimodal fusion. Several studies have proposed different approaches of multimodal fusion that would be discussed later. The diagrammatic representation in Figure 1 illustrates the process of multimodal stylometry.

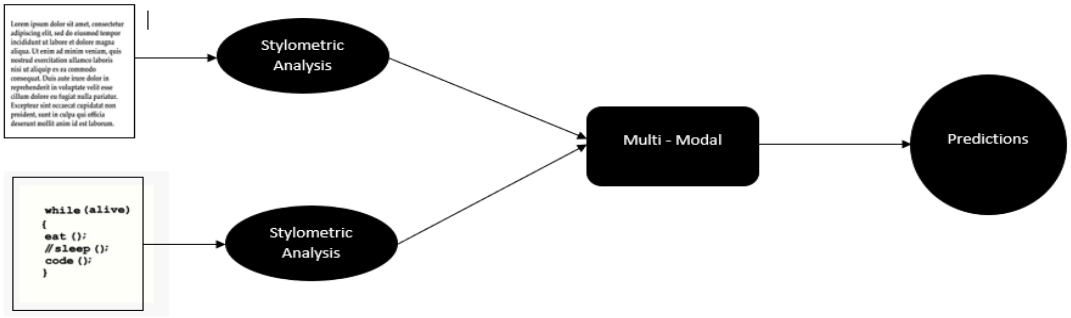

Figure 1: Multimodal Stylometry

1.5 Objectives of the Study

- Gather a dataset that comprises of text and source code written by the same author.
- Extract stylometric features from textual and source code data.
- Identify the best method of feature fusion for multimodal stylometry.
- Build a model from the extracted features for authorship identification.

- Can we use our methodology to distinguish between text and source code written by an AI chatbot (e.g., ChatGPT) and text and source code written by a human?

1.6 Significance of Study

Stylometry and by extension, multimodal stylometry carries profound implications across a multitude of domains, making it a field of immense relevance and impact. Its applications extend far and wide, encompassing forensic, security, digital humanities, literary analysis, academia, and cybersecurity. This study

delves into the heart of these applications, unraveling the potential and significance of multimodal stylometry in diverse areas of knowledge.

1.6.1 Forensic and Security Advancements

Multimodal stylometry plays a pivotal role in forensic and security endeavors. The ability to analyze non-linguistic features such as typing patterns, keystroke dynamics, and mouse movements holds the promise of identifying individuals engaged in computer-mediated communications. In contexts ranging from online harassment to cybercrime, this method offers a potent tool for tracking down wrongdoers. Multimodal stylometry goes beyond the boundaries of conventional linguistic analysis and single mode stylometry allowing for the identification of intricate patterns of style in both text and source code written by a hacker or cyberbully. It provides a holistic approach that surpasses the limitations of single-feature analysis, presenting an invaluable asset to those working in the fields of law enforcement and digital security.

1.6.2 Literary and Cultural Exploration

Beyond its forensic utility, multimodal stylometry adds a new dimension to digital humanities and literary analysis. Researchers find in it a means to explore shifts in an author's style, offering insights into the historical and cultural contexts that have left their mark on an author's work. By examining the evolution of vocabulary, sentence structure, and literary devices over time, scholars can unravel the intricacies of an author's creative journey. This approach enables a deeper understanding of how an author's style adapts and transforms in alignment with shifting cultural norms and evolving ideologies. It provides a lens through

which the narrative of literary and cultural change becomes clearer, fostering a deeper appreciation of the human creative process.

1.6.3 Academic Integrity

Multimodal stylometry also finds practical application in academia, especially within computer science and programming classes. In an academic setting, where students grapple with complex problems and submit both source code and written text, it can serve as a powerful tool for upholding academic integrity. Its capability to detect plagiarism and the use of large language models to complete class projects promotes honesty and ensures that students receive the recognition they deserve for their original work.

1.6.4 Cybersecurity and Ransomware Detection

In the realm of cybersecurity, multimodal stylometry assumes a critical role. The rising threat of ransomware attacks, which involve malicious payloads (source code) and ransom notes (text), can be more effectively countered through this approach. Multimodal stylometry equips experts with the means to uncover the identities of ransomware hackers, enhancing the chances of bringing them to justice. This capability represents a significant advancement in the ongoing battle against cyber threats.

In sum, this study not only investigates the potential of multimodal stylometry but also underscores its far-reaching significance, spanning from enhancing security and academic integrity to deepening our understanding of literary evolution and aiding in the identification of cybercriminals. The multifaceted implications of multimodal stylometry position it as a valuable tool with a broad spectrum of

applications, transcending traditional boundaries to reshape the way we analyze and understand various facets of human communication and behavior.

1.7 Research Contributions

The aim of this research is to show that multimodality in Stylometry improves the accuracy of identifying an author compared to unimodal stylometry. The introduction of multimodality in stylometry is a novel approach in stylometry and in this work we combine features from both text and source code. This approach promises to enhance the comprehensive understanding of authorship attributes by considering a broader spectrum of expressive elements. The research goes a step further by proposing an efficient feature fusion method, one that adeptly captures and consolidates information from both modalities. This fusion process extends the already existing early feature fusion by adding a second feature selection step after the fusion of features from the modalities we use (Text and Source Code).

1.8 Document Organization

The rest of our work is organized as follows.

In chapter 2, we review work that is related to our research. In Chapter 3, we present our proposed method. Chapter 4 gives a detailed evaluation of the results we obtained and finally, in chapter 5, we present our conclusions and propose future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The application of authorial discriminators to discern distinct authorial styles is a practice with a rich historical lineage, rooted in the fundamental premise that human behavior tends to follow patterns of consistency and habit. This approach, which forms the bedrock of stylometry, is not a recent development but an enduring concept. Its historical origins can be traced back to the mid-19th century when Augustus de Morgan, an English mathematician and logician, introduced this method. De Morgan's work extended beyond the theoretical realm, as he applied this approach to analyze the Pauline letters (Epistles) of the New Testament, marking an early and notable instance of authorship identification.

A follow up work was carried out by Thomas Corwin Mendenhall in 1887. During these early stages of stylometric analysis, the primary tool employed was the examination of frequency distributions, particularly in the form of histograms, capturing the variation in word lengths. These histograms emerged as a critical means of differentiation, shedding light on the distinctive writing styles of various authors.

The historical underpinnings of stylometry, characterized by the recognition of consistent authorial patterns, underscore the enduring nature of this approach. Over the centuries, it has evolved into a powerful and sophisticated field with a broad spectrum of applications, ranging from literary analysis to cybersecurity. The study of authorial styles continues to captivate researchers and scholars, reflecting the enduring relevance of this fundamental concept in the digital age.

The advent of the digital age, characterized by the widespread use of computers and the internet, has ushered in an era of unprecedented access to vast repositories of textual data. This exponential growth in the availability of text has, in turn, propelled the rapid expansion of the field of stylometry. The digital landscape is swarming with an abundance of textual content, spanning various genres, styles, and languages, and this wealth of data has become a fertile ground for stylometric exploration.

Furthermore, the proliferation of machine-generated text has added a new dimension to the field of stylometry. Coherent chatbots, now integrated into some social media platforms, possess the remarkable capability to deceive and mimic human communication. These advanced chatbots are equipped with artificial circadian rhythms, distinctive personas, and the ability to improvise by scouring the web for answers. For instance, certain text generators like SCIgen [1], originally designed to create fictitious research papers, has managed to hoodwink digital repositories, exposing the remarkable sophistication of these AI systems. Going

[1] https://pdos.csail.mit.edu/archive/scigen/

even further, chatbots such as ChatGPT have demonstrated the capacity to generate source code solutions for a wide array of user-proposed problems, underscoring the evolving and dynamic nature of human-machine interaction.

Despite the burgeoning interest in multimodality within other fields of research, the concept remains relatively nascent in the field of stylometry, and there is a dearth of established research papers in this specific domain. Consequently, in this chapter, we undertake a comprehensive review of the extensive body of work in the fields of text stylometry and code stylometry, encompassing various methods and techniques that have been developed to uncover authorship. We also explore the field of feature fusion which is a critical facet of multimodality. This is the process of combining the features from multiple modalities. Additionally, we delve into select studies that have ventured into multimodality in biometric, shedding light on their contributions and insights.

2.2 Corpus

In the field of Natural Language Processing (NLP) and, more specifically, Stylometry, the term "corpus" takes on a pivotal role. A corpus, by definition, encompasses all writings or works of a specific kind or on a particular subject, often referring to the complete literary output of an author. However, in the context of NLP and our current exploration into Stylometry, a corpus represents a diverse collection of texts or documents. This corpus functions as the fundamental dataset that is analyzed to make predictions or estimations.

The advent of the digital age and the meteoric rise of social media platforms, including Twitter and Facebook, have revolutionized the accessibility of vast corpora. The digital footprints left by individuals in the form of tweets, posts, and comments have ushered in a new era of data-rich resources for Stylometry and NLP. For instance, the CLEF initiative, the Conference and Labs of the Evaluation Forum [15], which is a self-organized body with the sole mission of promoting research, innovation and development of information access systems with an emphasis on multilingual and multimodal information with various levels of structure have since 2010 extracted tweets from twitter to build a corpus that is used by researches for author profiling tasks [16], [17], [18]

Beyond the realms of social media, numerous other sources of corpora have emerged to fuel the ever-growing demands of NLP research. Notable examples include the International Corpus for Learner English, crafted to explore the English writing of non-native English speakers [2], and genre-controlled corpora known as the British National Corpus [3].

It's worth noting that corpora can exist in a variety of formats, adapting to the needs of researchers and the demands of their projects. They may take the form of Extensible Markup Language (XML) structures, offering structured data for in-depth analysis [19], or they may present themselves in the more accessible and unadorned plain text format [20]. This flexibility ensures that corpora can be harnessed to suit the unique requirements of a wide array of NLP and Stylometry

investigations, providing a rich and versatile resource for the exploration of language, style, and authorship.

2.3 Text Stylometry

In the field of Stylometry, the most researched modality is text stylometry. Research in this area of stylometry encompasses a spectrum of intriguing findings and observations. Some studies have delved into sociolinguistic aspects, revealing the captivating notion that distinct groups of people, when communicating within a particular genre or using different languages, exhibit unique and identifiable linguistic characteristics [21]. In essence, the language itself becomes a canvas for stylistic variation, a notion that is both captivating and revealing.

Moreover, the exploration of text stylometry has unveiled a remarkable landscape of stylistic features that can be leveraged to determine the authorship of a written text. For instance, the presence of errors in writing, whether intentional or inadvertent, has emerged as a powerful discriminator for the native language of an author. These studies underscore the richness and complexity of text stylometry, where the idiosyncrasies of human expression come to the forefront, offering a trove of insights into authorial attribution.

Stylometry, according to Ramyaa et al [21], in the context of author attribution, assumes that an unconscious aspect exists to an author's style of writing that cannot be manipulated but possess distinctive and quantifiable features. These characteristic features an author possess should be frequent, salient, quantified easily and should be relatively immune to conscious control. Furthermore, these

features should be able to distinguish authors especially if they write in the same genre, on similar topics or even in the same period. Yet, within the field of stylometry, Ramyaa et al [21] has identified that one of the biggest problems is that there is no consensus as to what characteristic features, methodology or techniques that could be applied in standard research. This problem has been exhibited in most studies in stylometry were most of the experiments have been directed to different authors with different techniques and there has not necessarily been a comparison of results that demonstrates which features prove to be more representative or which techniques can be more effective.

Koppel et al. (2005) [2] approached the author profiling problem by showing that some stylistic text features (e.g. error in writing) could be used to determine the native language of an anonymous text. Their work illuminated the notion that certain stylistic characteristics of the errors made in a text could serve as potent determinants of the author's native language. To achieve this, they harnessed a diverse array of stylistic features, classifying them broadly into three categories:

1. **Function Words**: These encompass the frequently used, seemingly inconspicuous words that play essential roles in constructing sentences, such as articles, pronouns, and prepositions.
2. **Letter n-grams**: These involve the study of consecutive sequences of 'n' letters within words, which can offer insights into language-specific patterns.
3. **Errors and Idiosyncrasies**: This category delved into the intriguing world of linguistic errors and peculiarities.

Koppel et al. scrutinized various error types, automatically tagging them within documents. The four error types considered were:

- **Orthography**: Focusing on spelling errors, this category encompassed a range of issues, including missing letters and letter inversions.
- **Syntax**: Non-standard usages that deviate from conventional grammar, such as repeated words or missing words.
- **Neologisms**: The creation of neologisms and the study of parts-of-speech related to these innovations, like the playful "fantabulous."
- **Parts-of-Speech bigrams**: This category dealt with rare parts-of-speech bigrams, shedding light on unique linguistic patterns.

Their study leveraged the International Corpus for Learner English, a corpus designed for the study of English writing by non-native speakers. The authors in this corpus consisted of university students primarily in their 3rd or 4th years, all taking English as a second language class and typically in their 20s, demonstrating a similar proficiency in English. The nationalities represented included Russia, Czech Republic, Bulgaria, Spain, and France, with 258 authors considered for each language.

Each document in the corpus was represented by a numeric vector of length 1035, signifying the frequency of various features within the document. These features encompassed:

- 400 standard function words.

- 200 letter n-grams.
- 185 error types.
- 250 rare parts-of-speech bigrams.

The study employed a multi-class linear Support Vector Machine, employing a 10-fold cross-validation experiment to assess the accuracy of their methodology. The results were intriguing, showcasing that when all feature types were strategically juxtaposed, they achieved an accuracy of 80.2%. It's noteworthy that a significant portion of errors occurred among the three Slavic languages—Russian, Czech, and Bulgarian—indicating language-specific patterns that the methodology adeptly exploited.

The success of this methodology was highly dependent on the interaction of hundreds of features and as Koppel et al. (2005) [2] showed, there were several patterns that were unique to certain languages that they were easily able to exploit. For example, it was seen that for many authors in the Spanish corpus, there was a difficulty with doubling consonants (either they doubled unnecessarily as in *fullfill* or they omitted one of a double as in *effect*). This was also seen with a relatively huge number of the authors in the Czech corpus. This methodology also poses some question for future research (i) was method precise enough to handle a lot of different candidate native languages? (ii) How short can the body of text be and still permit accurate categorization?

Argamon et al. [23] explored the sociolinguistic observation that different groups of people speaking or writing in a genre and in a language use that language

differently. The main aim of the paper was to profile an author of a written text using a written text by the author. The profile dimensions that were being explored was gender, age, native language, and personality (Neuroticism). They identified content-based features and style-based features and applied Machine learning on the content-based features and style-based features independent of each other and combined them.

1. **Content-Based Features**: These features delved into the content of the written texts, exploring the linguistic patterns, word choices, and themes used by authors to convey their thoughts and ideas.
2. **Style-Based Features**: Style-based features probed the stylistic nuances present in the text, encompassing syntactic structures, punctuation usage, and other markers of an author's unique style and expression.

A novel feature set was introduced that naturally subsumes both functional and part-of-speech which has been known to be useful in linguistics. Systemic functional linguistics provided taxonomies describing meaningful distinctions among various function words and parts-of-speech. Three separate corpora were used to identify the profiles (age and gender shared the same corpus, but they were labelled differently):

- **Gender and Age Corpus**: This corpus was a comprehensive compilation of the complete writings of 19,320 blog authors. Its primary purpose was to enable profiling based on gender and age. Authors willingly self-reported

their age and gender, allowing for the creation of distinct categories such as Teens, Twenties+, and Thirties+.

- **Native Language Corpus**: Sourced from the International Corpus of Learner English, this corpus was curated to study the English writing of non-native speakers originating from various non-English-speaking countries, including Russia, Czech Republic, Bulgaria, Spain, and France. This sub-corpus comprised 258 authors from each language group, with any surpluses being randomly discarded.
- **Personality Corpus**: A unique facet of this exploration was the quest to profile authors based on their personality traits. To achieve this, the researchers tapped into essays written by psychology undergraduates at the University of Texas at Hendrix. The students were given the creative freedom to craft a "stream of consciousness" essay that mirrored their unfiltered thoughts and feelings during a 20-minute free-writing session. Furthermore, each writer completed a questionnaire assessing the "Big Five" personality dimensions. However, for this study, the spotlight was on neuroticism, a trait associated with worry and emotional instability.

To facilitate the classification tasks, the researchers strategically defined positive and negative examples. Positive examples included participants exhibiting neuroticism scores in the upper third, while negative examples consisted of those with scores in the lowest third. The research leveraged Bayesian Multinomial Regression, a probalisticaly well-founded multivariate logistic regression technique

known for its resilience against overfitting. This method had proven its effectiveness in a range of text classification and related problems.

The study rigorously examined the performance of their methodology through 10-fold cross-validation, yielding results that showed the potential of precise combinations of linguistic features and machine learning methods:

- Combining content-based and style-based features resulted in the most robust outcomes for age and gender profiling, achieving classification accuracies of 76.1% and 77.7%, respectively.
- Content-based features, when employed independently, emerged as the leaders in native language profiling, boasting a classification accuracy of 82.3%.
- Style-based features took the center stage in neuroticism profiling, achieving a classification accuracy of 65.7%.

This research demonstrated the immense potential of linguistic features and machine learning methods when combined. It illuminated the path toward the estimation of various profile aspects of an anonymous author, ushering in a new era of authorship exploration. In addition, it posed pivotal questions that open the doors to further research:

Can educational background and personality components be reliably extracted from texts, provided an appropriate training corpus and methodological framework?

To what extent do variations in genre and language influence the nature of models used in author profiling problems?

These intriguing questions beckon further exploration and hold the promise of improving research in the field of author profiling. The application of genre-controlled corpora and datasets featuring diverse languages promises to expand the horizons of author profiling research, offering a deeper understanding of the intricate web of linguistic authorship.

Koppel et al. (2002) [3] proposes a methodology showcasing how a genre-controlled corpus can be employed to automatically classify formally written texts according to the gender of the author. Their approach drew upon established techniques commonly used for text categorization and authorship attribution, delivering a fresh perspective to address this intriguing challenge.

The dataset at the core of their research consisted of 566 documents selected from the British National Corpus (BNC). What makes this corpus particularly unique is that it was constructed in such a way that no single author contributed more than three documents, ensuring a diverse and balanced representation. These documents varied in length, spanning from 554 to a substantial 61,199 words, with an average document size of 34,320 words.

One of the distinctive aspects of this methodology was the preprocessing phase. Instead of relying on a manually curated set of features, as is customary, the researchers began with an extensive collection of lexical and quasi-syntactic features. These features were chosen not based on their universality but rather on

their relevance to the specific topic at hand. This approach was groundbreaking in its departure from convention, seeking to explore the richness of language for author gender classification.

Each document was represented as a vector, comprising a total of 1,081 features. Within this feature set, 405 function words were included, each appearing at least once in the document. Additionally, a comprehensive list of part-of-speech (POS) n-grams was incorporated, utilizing the British National Corpus's tag set, which included 76 different parts of speech (e.g., PRP for prepositions, NNI for singular nouns, and so on). The top 500 most frequent ordered triples, the 100 most common ordered pairs, and all single tag features were considered. The utilization of POS n-grams was particularly ingenious, as it allowed the capture of deeper syntactic information.

However, the feature set was significantly streamlined using automated methods. These methods relied on iterative runs of a learning algorithm to eliminate features with low predictive power. In essence, this process can be described as a form of feature selection. Initially, a model was trained using all the available features. Subsequently, an automated procedure was employed to assign weights to each feature based on their contribution to the model's accuracy. Features with the highest weights were retained, while those with the lowest weights were systematically discarded.

The methodology's core mechanism revolved around identifying a linear separator between documents authored by male and female authors. This was accomplished

by assigning a weight vector (w) to each training document (x). The dot product of this weight vector (w) and the document vector (x) had to surpass a predetermined threshold (T) for the document to be classified as authored by a female writer. To compute these weights, the researchers employed a variant of the Exponential Gradient algorithm. This iterative process allowed for the continuous adjustment of weights based on a learning constant, which remained fixed at a value of 3 throughout the experiment. The weights were updated in such a way that they were increased for features that improved the accuracy and decreased for those that hindered it. While document vectors (x) could assume non-binary values, the score function (s(w, x)) was restricted to binary values (Balanced Winnow).

The process was further refined by randomly reordering the training samples and running another cycle, continuing until all training samples were classified correctly or until 100 consecutive cycles failed to yield an improvement in the classification accuracy. This iterative and adaptive approach showcased the methodology's ability to dynamically fine-tune its feature set and classification parameters.

Table 1 shows the results obtained after ten separate runs of 56-fold cross validation using a feature set that includes POS only, Function words only and both function words and POS. This experiment (& as seen in table 1) showed that using a combination of FW and POS yielded the best results across genres even though using more parameters (features) than constraints (documents) could have easily led to over-fitting during training thereby affecting the testing accuracy.

Domain	FW	POS	FWPOS
All	73.7 ± 0.86	70.5 ± 0.90	77.3 ± 0.79
Fiction	78.8 ± 1.1	77.1 ± 0.85	79.5 ± 1.1
Nonfiction	68.5 ± 1.3	67.2 ± 1.2	82.6 ± 0.99

Table 1: Results obtained after ten separate runs of 56-fold cross validation using a feature set that includes POS only, Function words only and both function words and POS [3]

Achieving higher accuracy across the board was indeed a commendable achievement, but it was not without its share of challenges, particularly when distinguishing between fiction and non-fiction. This divergence in content type was identified as a factor that adversely affected the results. To tackle this issue, the researchers implemented the Winnow algorithm, which adeptly harnessed subtle interdependencies between various features. In stark contrast, less nuanced learning methods like Naïve Bayes and Ripper proved less effective in mitigating this content-type disparity, leading to suboptimal classification performance.

The subsequent stage of the research sought to pinpoint the optimal number of features that would most significantly contribute to enhanced classification accuracy. To achieve this, for each model trained within the cross-validation trial, a selection process was implemented to identify 128 features deemed the most vital. The importance of a feature in each model was determined as the absolute value of its weight, multiplied by its average frequency in the training set. This curation resulted in a total of 256 selected features, half of which were assigned to each direction (e.g., male or female). The cross-validation process was then re-run exclusively using these carefully chosen features.

This selection process was iterated, each time halving the number of chosen features in each direction until a final set of only 8 features in each direction remained. The process aimed to pinpoint those with the most discriminative power, ultimately unveiling the most crucial linguistic distinctions between male and female authors within the realm of modern formal English articles.

The outcomes of this research displayed a compelling demonstration of the notable differences in the writing styles of male and female authors, specifically in the context of formal English articles. The Winnow-like algorithm, employed to harness these distinctions, achieved a classification accuracy of approximately 80%. This research underlines the existence of a marked divergence in writing styles between genders and effectively illustrates how certain selected features, along with their frequency distributions within the British National Corpus (BNC), significantly differ between male and female authors.

Furthermore, this study serves as an exemplar of the methodology adopted in contemporary text categorization research, with the primary differentiator being the selection of features. In this case, the focus was on content-independent features, shedding light on the efficacy of style-based categorization in addressing a wide array of text classification challenges. The approach showcased in this work holds the potential for success in other domains that necessitate style-based categorization techniques.

2.4 Feature Classification for Text Stylometry

In the field of stylometry, previous studies have set forth various feature classifications to quantitatively dissect an individual's writing style [21]. These classifications encompass an array of dimensions, each offering a unique lens through which an author's distinctive style can be examined. It's imperative to note that the focus of the current review lies not only in the identification of these stylometric features but also in understanding the computational resources and tools essential for their measurement [6]. Different feature categories demand varying levels of computational depth: Lexical and Character Features typically treat a text as a straightforward sequence of word tokens or individual characters. The computational requirements for measuring these features are relatively straightforward, as they primarily involve counting and analyzing the frequency and distribution of words and characters. In contrast, syntactic and semantic features require more profound linguistic analysis. These features necessitate the use of advanced linguistic tools and resources, such as syntactic parsers and semantic analyzers. They delve into the structural and meaning-based aspects of language, demanding a higher level of computational complexity. There are also features tailored to specific text domains or languages that can only be defined and measured within those defined contexts. For instance, in an HTML-based corpus, features like font color counts or font size counts might be crucial [24]. This highlights the need for domain-specific expertise and resources to capture the nuances unique to a particular application.

The following tables provide a comprehensive overview of the stylometric feature types, offering insight into their key characteristics and the computational tools and resources essential for their measurement. These features collectively serve as the building blocks for unraveling the intricacies of stylometry and offer a rich foundation for advancing the field.

2.4.1 Lexical Features

This category dissects the text as a sequence of word tokens. Lexical features often consider the author's vocabulary richness, word frequency, and the distribution of words throughout the text. They provide valuable insights into an author's diction and linguistic choices.

Features	**Required Tools and Resources**
Token-based (word length, sentence length etc.)	Tokenizer, [Sentence Splitter]
Vocabulary richness	Tokenizer
Word frequencies	Tokenizer, [Stemmer, Lemmatizer]
Word n-gram	Tokenizer
Errors	Tokenizer, Orthographic spell checker

Table 2: Stylometric lexical features and the computational tools & resources required to measure them [6]

2.4.3 Character Features

On the other hand, character features zoom in even further, examining the text as a sequence of individual characters. They delve into aspects like character frequency, character combinations, and character-level patterns within the text. These features uncover nuances in an author's textual fingerprint.

Features	**Required Tools and Resources**
Character types (letters, digits, etc.)	Character dictionary
Character n-grams (fixed – length)	-
Character n-grams (variable – length)	Feature selector
Compression methods	Text compression tool

Table 3: Stylometric character features and the computational tools & resources required to measure them [6]

2.4.4 Syntactic Features

Unlike the previous two categories, syntactic features require a deeper level of linguistic analysis. They delve into the sentence structure, grammatical elements, and the arrangement of words. Syntactic features explore how an author constructs sentences, their use of punctuation, and grammatical idiosyncrasies.

Features	**Required Tools and Resources**
Part-of-Speech	Tokenizer, Sentence Splitter, POS tagger
Chunks	Tokenizer, Sentence splitter, [POS tagger], text chunker
Sentence and phrase structure	Tokenizer, Sentence Splitter, POS tagger, Text chunker, Partial parser
Rewrite rules frequencies	Tokenizer, Sentence splitter, POS tagger, Text chunker, Full parser
Errors	Tokenizer, Syntactic spell checker, Sentence splitter

Table 4: Stylometric syntactic features and the computational tools & resources required to measure them [6]

2.4.5 Semantic Features

This category of features scrutinizes the author's use of semantics, examining how words and phrases are employed to convey ideas. Semantic features tap into the deeper layers of an author's style, unveiling their unique ways of expressing thoughts and ideas.

Features	**Required Tools and Resources**
Synonyms	Tokenizer, [POS tagger], Thesaurus
Semantic Dependencies	Tokenizer, Sentence splitter, POS tagger, Text Chunker, Partial parser, Semantic parser
Functional	Tokenizer, Sentence splitter, POS tagger, Specialized dictionaries

Table 5: Stylometric semantic features and the computational tools & resources required to measure them [6]

2.4.6 Application – Specific Features

These features are tailored to a particular text domain or language domain, with a clear focus on characteristics that hold relevance within those specific contexts.

Features	Required Tools and Resources
Structural	HTML parser, Specialized parsers
Content – specific	Tokenizer, [Stemmer, Lemmatizer], Specialized dictionaries
Language – Specific	Tokenizer, [Stemmer, Lemmatizer], Specialized dictionaries

Table 6: Stylometric Application – specific features and the computational tools & resources required to measure them [6]

2.5 Code stylometry

The importance of code stylometry is something that is heavily disputed especially among underground programmers or programmers on the dark web. Certain programmers decide to hide their identity for several reasons (a programmer who does not want their employer to know about their side gigs, the creator of bitcoin or they may live in a country that prohibits some type of software). A poignant illustration of the stakes involved can be found in the case of an Iranian programmer in 2012. This individual faced a dire sentence and capital punishment, for their involvement in developing a seemingly innocuous photo sharing software that was ultimately used on pornographic websites [25]. This tragic event underscores the potential life-and-death consequences that can emerge from code authorship within certain jurisdictions.

However, on the other hand, source code attribution might be very useful in forensics (ghostwriting detection, plagiarism, or copyright investigation). Furthermore, code stylometry can also help in detecting the identity of authors of malware. In this section, we present an overview of work done in code stylometry especially as it relates to our research.

Caliskan-Islam et al [25] presents research that showed that Abstract Syntax Trees (ASTs) carry authorial ‘fingerprints. Their work not only achieved good accuracy but also overcame limitations that were encountered in previous studies [26]. This milestone in code stylometry yielded a 97% accuracy on a small sample set of 30 programmers. Significantly, their success was achieved without relying on programmers' comments, which had been a common practice, and with a reduced dependency on extensive training data [27], [28]. Furthermore, Caliskan-Islam's research attained an accuracy of 92.83% on a larger dataset comprising 1600 programmers, demonstrating the scalability and effectiveness of their approach. Central to their contribution was the astute utilization of syntactic features extracted from Abstract Syntax Trees, which provided a syntactic style representation for code stylometry.

The corpus consisted of code written by 250 different programmers for nine (9) different problems written in C and C++. The comments were taken out from the code samples to identify authors on purely just their coding style. During the feature extraction stage, the use of unigram term frequency and TF/IDF measures and the diversity of individual terms in the code yielded a large and spare feature

vector with over 120,000 features. The features used can be classified into Lexical, Layout and Syntactic (extracted using Abstract Syntax Trees) features.

Due to the number of features extracted from the source code, the curse of dimensionality was very evident which meant that the feature set had to be reduced. A feature selection process was introduced, leveraging WEKA's information gain criterion [29]. This criterion, essentially, assesses the individual worth of each feature concerning its contribution to the class, as articulated in Equation 1:

$$IG(A, M_i) = H(A) - H(A|M_i) \qquad (1)$$

Where:

A is the class corresponding to an author.

H is the Shannon entropy

M_i *is the ith feature of the dataset.*

The Feature set was ranked based on the extent of information they supplied towards the outcome or class. Those features demonstrating non-zero information gain were selectively retained and coined as IG-CSFS features. In a bid to validate the robustness of this approach, the dataset was bifurcated into two subsets, and an intriguing consistency emerged: the features with non-zero information gain exhibited striking similarity across both sets, underscoring the reliability of the feature selection process.

For the classification task, a Random Forest classifier was used as the learning algorithm, and a 10-fold cross-validation methodology was employed. The resulting classification model was subjected to a comprehensive assessment against another dataset, encompassing 250 programmers and nine distinct solutions spanning various years of the Google Code Jam (GCJ) [2] competition.

The results can be seen in the table below.

A = #programmers, F = max #problems completed		
N = #problems included in dataset (N ≤ F)		
A = 250 from 2014	A = 250 from 2012	A = 250 all years
F = 9 from 2014	F = 9 from 2014	F ≥ 9 all years
N = 9	N = 9	N = 9
Average accuracy after 10 iterations with IG-CSFS features		
95.07%	**96.83%**	**98.04%**

Table 7: Validation Experiments for Caliskan-Islam et al [25]

One of the phenomena that was explored by the researchers was to see the effect of code obfuscation on the accuracy of their model. The table below shows the results obtained after the codes were passed through an obfuscation software.

Obfuscator	**Programmers**	**Language**	**Results without Obfuscation**	**Results with obfuscation**
Stunnix	20	C++	98.89%	100.00%
Stunnix	20	C++	98.89%	98.89%
Tigress	20	C	93.65%	58.33%
Tigress	20	C	95.91%	67.22%

Table 8: Effects of Obfuscation on De-anonymization [25]

The culmination of this study goes beyond mere standalone observations. It extends its impact by situating the results within the broader landscape of code stylometry. The researchers thoughtfully benchmarked their findings by comparing

[2] https://code.google.com/codejam,

them to the outcomes of prior studies in this field, thereby facilitating a comprehensive understanding of the research's context and its contribution to the existing body of knowledge.

By juxtaposing these results with those of past studies, this research endeavor establishes a crucial link to the existing literature, highlighting both the consistencies and novel insights that have emerged from this investigation.

This result is presented in the table below.

Related Work	**Number of Programmers**	**Results**
Pellin [28]	2	73.00%
MacDonell et al. [29]	7	88.00%
Frantzeskou et al. [26]	8	100.00%
Burrows et al. [30]	10	76.78%
Elenbogen and Seliya [31]	12	74.70%
Kothari et al. [32]	12	76.00%
Lange and Mancordis [33]	20	75.00%
Krsul and Spafford [34]	29	73.00%
Frantzeskou et al. [26]	30	96.90%
Ding and Samadzadeh [35]	46	67.20%
Caliskan – Islam et al. [12]	**8**	**100.00%**
Caliskan – Islam et al. [12]	**35**	**100.00%**
Caliskan – Islam et al. [12]	**250**	**98.04%**
Caliskan – Islam et al. [12]	**1600**	**92.83%**

Table 9: Comparison of results between previous studies and Caliskan et al. [25]

Dauber et al. [8], explored the problem of authorship attribution by using code obtained from open-source systems (GitHub) to ascertain how contributions can be attributed to either the individual authors or based on the contributing accounts. The source codes used were fragments and not necessarily complete programs. The aim of this work was to see if they could identify authors who have contributed

to a complete work by identifying authorship styles in the different fragments of the code.

They revealed that when previous methods [25] were applied to these code fragments, as opposed to complete programs, the accuracy of attribution peaked at a modest 50% to 60% at best. This highlighted the challenge of dissecting and identifying authorship patterns within code fragments, which are often more compact and exhibit less comprehensive authorship indicators.

A key innovation introduced in this study involved the use of calibration curves to assess the trustworthiness of attributions made for unknown and previously unencountered authors. These curves leveraged the output probabilities, effectively assessing the confidence levels of the classifiers. The critical threshold delineated below the calibration curve was used to categorize an author as either within the known and suspect set (Closed World) or outside of it (Open World). This sophisticated approach brought a new dimension to authorship attribution in code, acknowledging the dynamic and ever-evolving landscape of collaborative coding projects.

In addition, two significant modifications were incorporated into the method inspired by Caliskan-Islam et al. [25] to tailor it for the specific context of code fragments:

1. A form of ensembling whereby they ensembled outputs of multiple linked samples for the classifier instead of ensembling the outputs of different

classifiers on the same sample. This showed that accuracy can be greatly improved if they were able to link several fragments to the same unknown author.

2. The use of calibration curves to determine if an attribution can be trusted for a given fragment or a set of code fragments.

Fig 2 shows the methodology of this work.

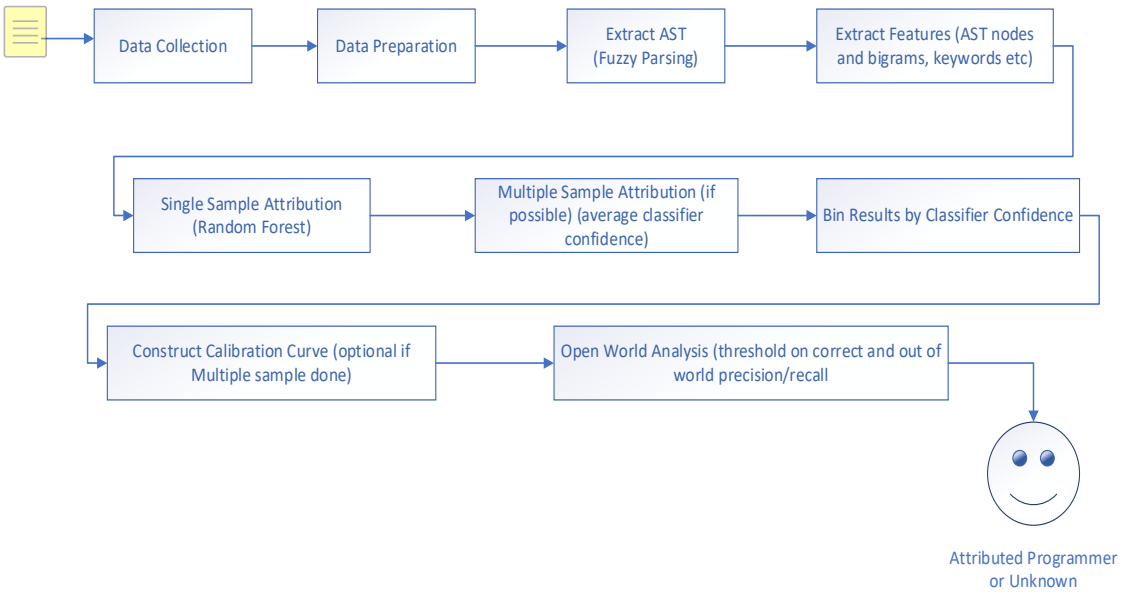

Figure 2: Authorship Attribution methodology for Dauber et al. [8]

In this study, data was gathered from a repository of 1649 C++ projects hosted on GitHub[3], involving contributions from 1178 programmers (also referred to as contributors). After data preprocessing was completed (programmers with at least 150 fragments that was at least one line of code), 104 programmers were left. An important aspect of this study, worth highlighting, is the retention of comments within the code, a departure from previous research like Caliskan-Islam et al. [25],

[3] https://github.com/

which often stripped comments. This unique approach brought this study closer to the real-world coding scenarios, where comments can carry valuable contextual information.

The feature set utilized in this investigation was extracted using Abstract Syntax Trees, a technique akin to Caliskan-Islam et al. [25]. It encompassed word unigrams, API symbols, and keywords, revealing the intricate linguistic characteristics imprinted in the code. The results of this meticulous research endeavor yielded a series of intriguing findings:

1. **Authorship Attribution and the Closed World**: The study revealed that authorship attribution was markedly more manageable when all programmers were treated within the suspect set, adhering to the Closed World paradigm.
2. **The Influence of Sample Size**: An interesting observation was that increasing the number of code samples significantly contributed to the accuracy of the attribution method.
3. **Comments and Personalized Function Declarations**: Code samples containing plaintext, such as comments, and those featuring highly personalized function declarations, presented distinct characteristics that were easily attributable to their respective authors. Interestingly, these features could potentially be exploited to deceive the classifiers, underscoring the intricate interplay between code style and authorship attribution.

4. **Anonymity for Short Codes**: The research revealed that authors of short code snippets could maintain a degree of anonymity for a limited period if they took specific precautions to obfuscate their code. However, this anonymity was not guaranteed.
5. **The Evolving Landscape of Authorship Attribution**: As advances continue to be made in authorship attribution and as more features are extracted from code, it is anticipated that programmers' ability to remain anonymous online will diminish. The increasing sophistication of attribution techniques and the expanding feature set availability are set to challenge the boundaries of online anonymity in the coding world.

This study stands as a significant milestone in the field of code stylometry, offering valuable insights into the challenges and potential solutions in attributing code to its authors. By embracing the complexity of real-world coding scenarios and retaining comment data, this research offers a more realistic depiction of the coding landscape. It opens exciting possibilities for enhancing attribution methods and understanding the evolving dynamics of code attribution in a digital age.

2.6 Feature Classification for Code Stylometry

In the field of code stylometry, a comprehensive understanding of the features employed to discern the nuances of an author's programming style can be classified into three: Lexical, Layout, and Syntactic features.

2.6.1 Lexical Features

Lexical features derive their essence from the source code itself and are gauged by counting specific elements, tokens, keywords, or the length of functions. These features delve into the lexical richness of the code, shedding light on the author's choice of expressions and the distribution of linguistic elements within the codebase. Lexical features are instrumental in unveiling an author's programming vocabulary and the frequency of constructs.

Feature	Definition
WordUnigramTF	Term frequency of word unigrams in source code
ln(numkeyword/length)	Log of the number of occurrences of keyword divided by file length in characters, where keyword is one of do, else-if, if, else, switch, for or while
ln(numTernary/length)	Log of the number of ternary operators divided by file length in characters
ln(numTokens/length)	Log of the number of word tokens divided by file length in characters
ln(numComments/length)	Log of the number of comments divided by file length in characters
ln(numLiterals/length)	Log of the number of strings, character, and numeric literals divided by file length in characters
ln(numKeywords/length)	Log of the number of unique keywords used divided by file length in characters
ln(numFunctions/length)	Log of the number of functions divided by file length in characters
ln(numMacros/length)	Log of the number of preprocessor directives divided by file length in characters
nestingDepth	Highest degree to which control statements and loops are nested within each other
avgParams	The average number of parameters among all functions
stdDevNumParams	The standard deviation of the number of parameters among all functions
avgLineLength	The average length of each line
stdDevLineLength	The standard deviation of the character lengths of each line
branchingFactor	Branching factor of the tree formed by converting code blocks of files into nodes

avgParams	The average number of parameters among all functions.
stdDevNumParams	The standard deviation of the number of parameters among all functions
AvgLineLength	The average length of each line
stdDevLineLength	The standard deviation of the character lengths of each line

Table 10: Lexical Features for Code Stylometry and their definitions

2.6.2 Layout Features

Layout features are centered around the spatial arrangement of code, notably code indentation. This category explores the structural aspects of code formatting, providing insights into the author's coding style. For instance, a typical layout feature could involve calculating the ratio of whitespace characters at the beginning of a line relative to the overall file size. By assessing code indentation patterns, layout features decode an author's approach to structuring their code, reflecting their stylistic preferences.

Feature	Definition
ln(numTabs/length)	Log of the number of tab characters divided by file length in characters
ln(numSpaces/length)	Log of the number of space characters divided by file length in characters
ln(numEmptyLines/length)	Log of the number of empty lines divided by file length in characters, excluding leading and trailing lines between lines of text
whiteSpaceRatio	The ratio between the number of whitespace characters (spaces, tabs, and newlines) and non-whitespace characters
newLineBeforeOpenBrace	A boolean representing whether most code-block braces are preceded by a newline character
tabsLeadLines	A boolean representing whether most indented lines begin with spaces or tabs

Table 11: Layout Features for Code Stylometry and their definitions

2.6.3 Syntactic Features

Syntactic features dive into the structural intricacies of the source code and are rooted in language-dependent abstract syntax trees (ASTs). Abstract syntax trees represent a hierarchical, tree-like structure that encapsulates the syntactic structure of a block or segment of source code. These trees break down the code into its constituent elements, facilitating a granular examination of the code's structure. Syntactic features extracted from ASTs unlock a deeper level of insight into the author's coding style, uncovering their approach to code organization, the arrangement of control flow constructs, and the interaction of code components.

Feature	Definition
MaxDepthASTNode	Maximum depth of an AST node
ASTNodeBigramsTF	Term frequency AST node bigrams
ASTNodeTypesTF	Term Frequency of 58 possible AST node type excluding leaves
ASTNodeTypesTFIDF	Term frequency inverse document frequency of 58 possible AST node type excluding leaves
ASTNodeTypeAvgDep	Average depth of 58 possible AST node types excluding leaves
cppKeywords	Term frequency of 84 C++ keywords
CodeInASTLeavesTF	Term frequency of code unigrams in AST leaves
CodeInASTLeavesTFIDF	Term frequency inverse document frequency of code unigrams in AST leaves
CodeInASTLeavesAvgDep	Average depth of code unigrams in AST leaves

Table 12: Syntactic Features (Extracted from ASTs) for Code Stylometry and their definitions.

Therefore, syntactic features are extracted from the leaves and nodes of the tree. Consider the pseudocode below.

x = y + 3

The AST representation for the above pseudocode can be seen in Figure 3

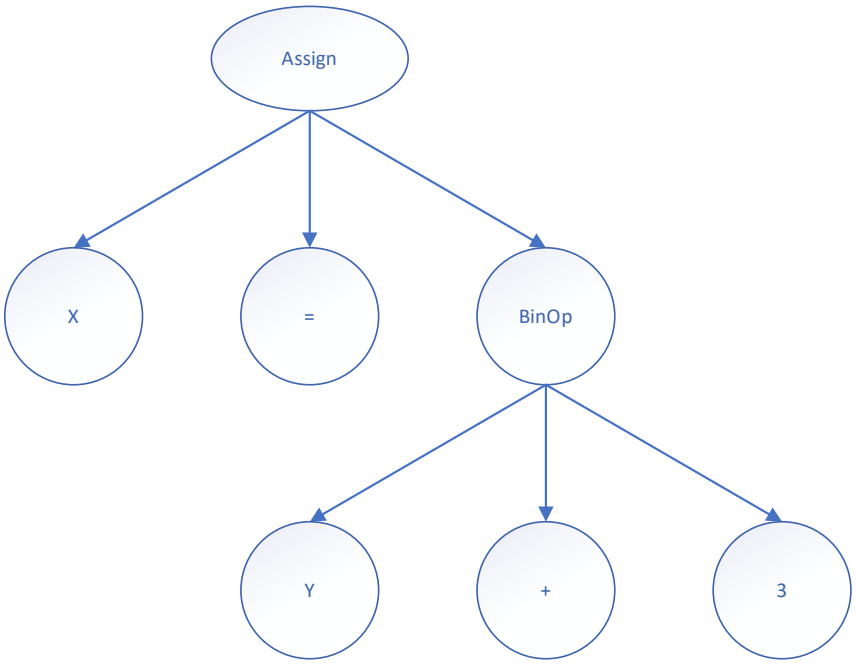

Figure 3: AST representation of the pseudocode x = y+3

In essence, Lexical, Layout, and Syntactic features are the fundamental building blocks of code stylometry. They collectively enable the in-depth analysis of source code, empowering researchers to discern the author's unique fingerprints woven into the fabric of their code. By combining these feature categories, code stylometry ventures beyond the surface of code repositories, offering a profound understanding of programming styles, syntax choices, and code organization [25].

2.7 Feature Fusion

Feature Fusion is a vital component of multimodality, and it entails combining the information drawn from various modalities into a cohesive framework for making predictions and inferences. In this section, we look at the various feature fusion techniques used in multimodality. In essence, feature fusion represents the synergy of information extracted from a medley of sources or modalities.

Multimodality boasts multiple tiers of feature fusion, catering to distinct research domains and objectives. In the field of biometrics, for instance, feature fusion can manifest at various levels:

2.7.1 Feature Level Fusion

The process of fusing features from multiple modalities plays a pivotal role in multimodality. It encompasses the collection and preprocessing of datasets, followed by feature extraction. With feature level fusion, these features are then merged through concatenation, and the resulting feature set is utilized for classification purposes. This approach aligns the data at its most granular level, fostering a direct juxtaposition of feature vectors. The resulting feature fusion reflects the diverse characteristics of each modality, creating a richer, multidimensional representation.

Some research has shown that feature level fusion when compared with other fusion levels, is the most effective [30] [31], . Fusion at feature level provides

more distinctive information because of the high dimensional space between the impostor and the genuine samples.

For instance, Hezil et al. [32] conducted a remarkable study involving feature level fusion of ear and palm print data. Their findings demonstrated a notable increase in recognition rates when compared to unimodal biometrics. Additionally, Haghighat et al. [30] leveraged Discriminant Correlation Analysis (DCA) in their feature level fusion approach, effectively maximizing pairwise correlations while diminishing between-class correlations. These endeavors underscore the substantial potential of feature level fusion in enhancing biometric recognition systems.

2.7.2 Comparison Score Level Fusion

Comparison scores fusion (known previously as “matching scores”) is done by combining results obtained from many modalities. These results are obtained by calculating similarity or the distance measurement for the input sample and the template from the dataset. T-norms functions can be used to merge the input scores usually using max, min or some fuzzy logic combinations of the results.. Hanmandlu et al. [33] conducted an extensive evaluation of score level combination approaches, emphasizing the use of T-norms. In a similar vein, Ross et al. [34] explored score level fusion employing face, fingerprint, and hand geometry in the context of a multimodal biometric system. The simplicity and versatility of fusion at the comparison score level make it adaptable to a multitude of modalities.

2.7.3 Decision Level Fusion

Decision level fusion involves the integration of individual decisions or outcomes derived from each modality. This fusion level requires a framework for aggregating the decisions and arriving at a final consensus or inference. Decision level fusion is particularly potent in scenarios where modalities offer complementary insights. Following the preprocessing of each modality, an individual classification is obtained from each source. These classification results are subsequently fused using logical operations such as min, max, or min-max, which can be further enhanced through the incorporation of fuzzy logic. Score level and decision level bear some similarities since in both cases a decision is made after results have been obtained from the individual modalities.

Rajasekar et al. [35] have showcased decision and score level fusion techniques, showcasing how decision-making can be refined through the application of an optimized fuzzy genetic algorithm.

2.7.4 Hybrid Fusion

This is a combination of feature level fusion and decision level fusion. With the hybrid fusion, data from each modality is preprocessed, features extracted and classified separately. Also, the features are combined in a separate experiment and classified. The results obtained from the individual modalities and the combination of features from the modalities are now combined at the decision level using some form of logical operation.

Wang et al. [36] aimed to maximize information gain by combining features derived from feature level fusion with features from individual modalities. This

hybrid approach reflects the quest for superior biometric recognition performance, leveraging the strengths of both feature and decision level fusion strategies to achieve the desired outcomes.

In multimodality research, feature fusion emerges as an indispensable tool for unlocking the potential hidden within the convergence of diverse modalities.

2.8 Multimodal Biometrics

This section provides a succinct yet comprehensive overview of multimodal biometrics, as it aligns closely with the focal point of our research. In biometrics research, a system is deemed biometric when it possesses the capability to automatically identify and authenticate individuals based on their unique physiological or behavioral traits. These traits encompass a broad spectrum, including but not limited to facial features, retinal patterns, vein structures, speech patterns, ear design, nail bed characteristics, keystroke dynamics, and fingerprints. While fingerprint recognition stands out as one of the most frequently employed biometric traits due to its prevalence and reliability, the iris emerges as a standout contender in the terms of accuracy. The iris's distinctiveness and consistency render it a powerhouse in the biometric landscape.

While unimodal biometric techniques can often deliver high accuracy, the synergy of multiple modalities, encapsulated within multimodal biometrics, offers a range of compelling advantages. Multimodal biometrics, with its fusion of various biometric traits, excels in terms of resilience against spoofing attacks and enhances the overall capabilities of biometric recognition systems [37].

Rajasekar et al. [38] explored a multimodal biometric technique that improved accuracy and recognition rate for a biometric system in a smart city. The technique used is based in score-level fusion. This work focused on using two modalities (fingerprint and iris) with a score-level fusion technique which also consisted of an enhanced optimized fuzzy genetic algorithm (OFGA). The fig 3 below shows a flow of the proposed method.

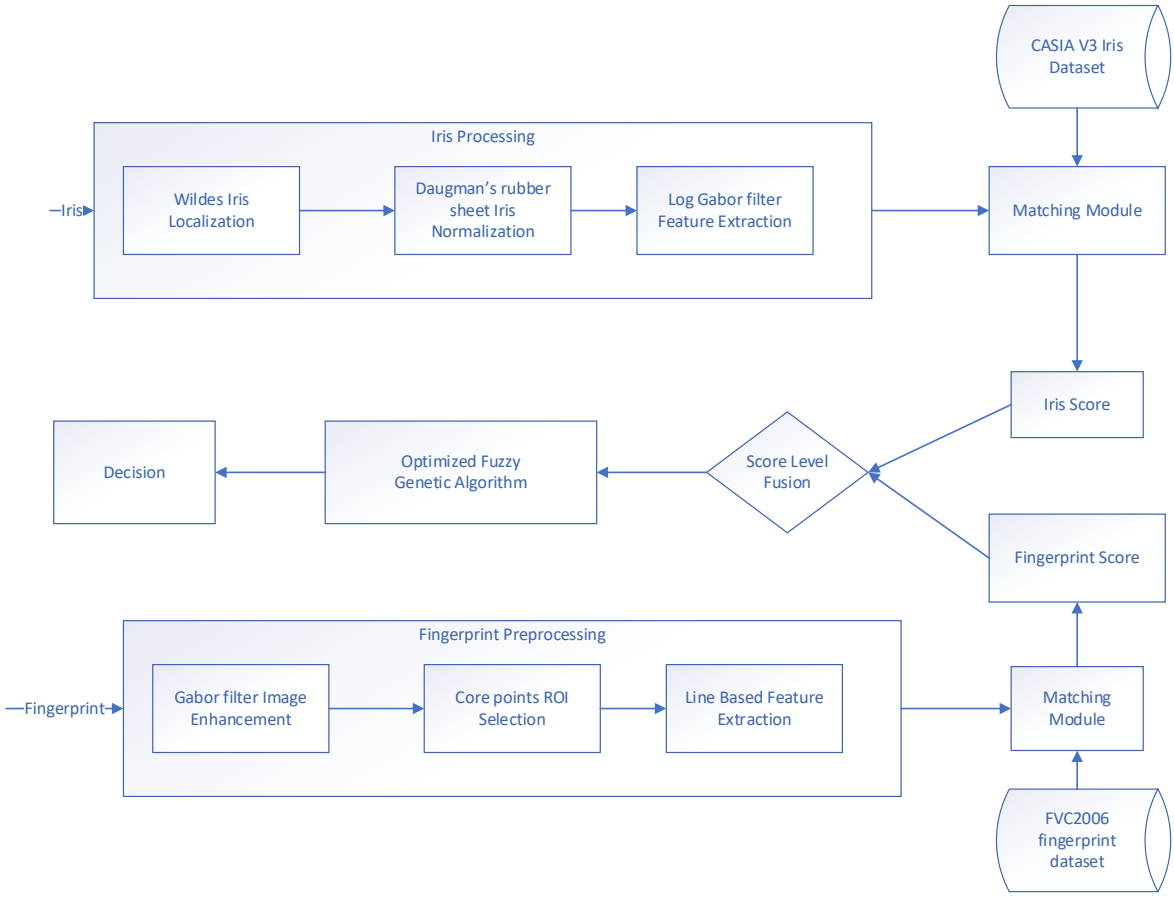

Figure 4: Process flow and methodology for Rajasekar et al. [38]

The data used were the CASIA Iris V3 dataset (iris) and the FVC2006 fingerprint dataset (fingerprints). Their techniques used the OFGA to improve the recognition of fingerprints and Iris data for biometrics. The crux of this methodology involved

the extraction of morphological features from both fingerprints and irises. They quantified the matching rates, denoted as MR_f for fingerprints and MR_i for irises, as depicted in Equations 2 and 3:

$$MR_f = w_f m_f \qquad (2)$$

$$MR_i = w_i m_i \qquad (3)$$

Where, MR_f and MR_i represent the matching rates for fingerprints and iris, with w_f and w_i signifying the corresponding weights, while m_f and m_i denote the matching scores for fingerprints and irises, respectively.

To fuse the matching scores obtained from these diverse modalities, they employed the weighted sum rule, as defined in Equation 4:

$$M_s = w_i m_i + w_f m_f \qquad (4)$$

The OFGA method proposed is a stochastic optimization method that combines fuzzy approach and a genetics algorithm where mutation and crossover are incorporated to minimize convergence. The primary objective of the OFGA is to minimize the weights, as exemplified in Equation 5:

$$(w_{i,} w_f) \qquad (5)$$

Where obj(Z) symbolizes the objective function of feature Z, and the minimization of the weight vector w is the overarching goal.

The steps integral to the genetic algorithm implementation are as follows:

1. Initialization: In this inaugural phase, the genetic algorithm generates an initial population with randomized parameters.

2. Fitness Function: For their approach, the Equal Error Rate (EER) and accuracy was considered as the fitness functions of the OFGA. In this context, higher accuracy and lower EER values signify enhanced biometric recognition. Each member of the population is assigned a fitness value closely linked to these fitness functions, thus reflecting the importance of each function. These fitness values play a pivotal role in the global convergence achieved at the culmination of the mission.

3. Fuzzy Clustering Approach (Selection): This crucial step entails the allocation of each data point into specific clusters. For every population *y* = (y_1, y_2, y_3,, y_n), rules were defined that segregate the data into distinct *C* = (c_1,c_2, c_3, , c_m), minimizing data feature O_x, where the degree of fuzziness is constrained by the factor x. The partition matrix is denoted as W = w_{ij}, indicating that element y_i belongs to C_j, as elucidated in Equation 6:

$$arg\ \_min\ _C = \sum_{i=1}^{i=n} \sum_{j=1}^{i=m} w_{ij}|y_i - c_j|^{\ 2} \qquad (6)$$

Where.

$$w_{ij} = \frac{1}{\sum_{k=1}^{m} \left(\frac{y_i - c_j}{y_i - c_k}\right)^{\frac{2}{m-1}}} \qquad (7)$$

4. Crossover and Mutation: The corresponding correlations and probabilities for the crossover and mutation operators are thoughtfully applied to the global solution, as shown in Equation 8:

$$p_i^j = p_i^{j-1} \times rand(y_i^{j-1} - x_i^{j-1}) \qquad (8)$$

The results obtained can be seen below.

N	TP	FN	FP	TN	FAR (%)	FRR (%)	TPR (%)	TNR (%)	Precision	Accuracy (%)
100	100	0	0	100	0	0	100	100	100	100
300	300	0	0	300	0	0	100	100	100	100
500	499	1	1	499	0.2	0.2	99.8	99.8	99.8	99.8
800	797	3	2	798	0.25	0.38	99.63	99.75	99.75	99.63
1000	997	3	3	997	0.3	0.3	99.7	99.7	99.7	99.7
1300	1296	4	4	1296	0.31	0.31	99.69	99.69	99.69	99.69
1500	1495	5	5	1495	0.33	0.33	99.67	99.67	99.67	99.67
1800	1792	8	8	1792	0.44	0.44	99.56	99.56	99.56	99.56
2100	2089	11	12	2088	0.57	0.52	99.48	99.43	99.43	99.48

Table 13: Performance evaluation without OFGA [38]

N	TP	FN	FP	TN	FAR (%)	FRR (%)	TPR (%)	TNR (%)	Precision	Accuracy (%)
100	100	0	0	100	0	0	100	100	100	100
300	300	0	0	300	0	0	100	100	100	100
500	500	0	0	500	0	0	100	100	100	100
800	798	2	1	799	0.13	0.25	99.75	99.88	99.87	99.81
1000	998	2	1	999	0.18	0.18	99.8	99.9	99.9	99.85
1300	1296	4	2	1298	0.15	0.31	99.69	99.85	99.85	99.77
1500	1497	3	1	1499	0.07	0.02	99.8	99.93	99.93	99.87
1800	1792	7	5	1795	0.28	0.39	99.61	99.72	99.72	99.67
2100	2090	10	8	2092	0.38	0.48	99.52	99.62	99.62	99.57

Table 14: Performance evaluation with OFGA [38]

Modalities	FAR (%)	FRR (%)	TPR (%)	TNR (%)	Accuracy (%)	EER (%)	Precision
Iris and Fingerprint without OFGA	0.26	0.27	99.73	99.74	99.74	0.33	99.73
Iris and Fingerprint with OFGA	0.13	0.20	99.79	99.88	99.83	0.18	99.88

Table 15: Average performance evaluation [38]

Approaches	FAR (%)	FRR (%)	Accuracy (%)	EER (%)
Gavisddappa et al. [45]	9.87	11.89	97	0.23
Jagadiswary et al. [46]	0.01	0.27	87	0.37
Vidya & Chandrause [47]	0.32	0.33	91	0.32
Yang et al. [48]	0.38	0.27	90	0.33
Malarvizhi et al. [49]	0.58	0.02	96	0.22
Selwal et al. [50]	3.30	3.39	97	0.20
Rajasekar et al. [42]	0.13	0.20	99.83	0.18

Table 16: Comparison of existing approaches with Rajasekar et al. [38]

Where.

FP = False Positives

TN = True Negatives

FN = False Negatives

TP = True Positives

FAR = False Acceptance Rate, the likelihood of a system incorrectly accepting a nonregistered or unauthorized user.

FRR = False Rejection Rate, the likelihood of a system incorrectly rejecting a nonregistered or unauthorized user.

TRR = True positive rate, the likelihood of a device authorizing the registered user. It is also known as sensitivity.

TNR = True Negative Rate, likelihood of the authorized user being approved by a system. It is otherwise defined as recall or specificity.

Accuracy = registered users permitted at a rate proportional to the number of attempts they made.

ERR = Equal error rate, the rate at which FAR is equal to FRR.

Precision = the ratio of positive instances found among all positives mentioned. It is otherwise denoted as a positive predictive value.

This research explores a novel approach for multimodal biometrics in a smart city that uses fingerprint and Iris. The features extracted from the modalities are given as input into a score-level fusion step. The output from this is fed into an optimized fuzzy genetic algorithm (OFGA) which improves the accuracy of biometric recognition. The proposed approach yielded an accuracy of approximately 99.89% and an EER of approximately 0.18%.

Wang et al. [36] proposes a two-channel convolutional neural network (CNN) fusion framework which can be seen in fig 4, the fusion also occurs at the feature level. The modalities used were finger vein and facial recognition which were in the form of images. The datasets used were SDUMLA-FV (Finger vein), CASIA

WebFace (Face) and the USM-FV (Finger vein) datasets.

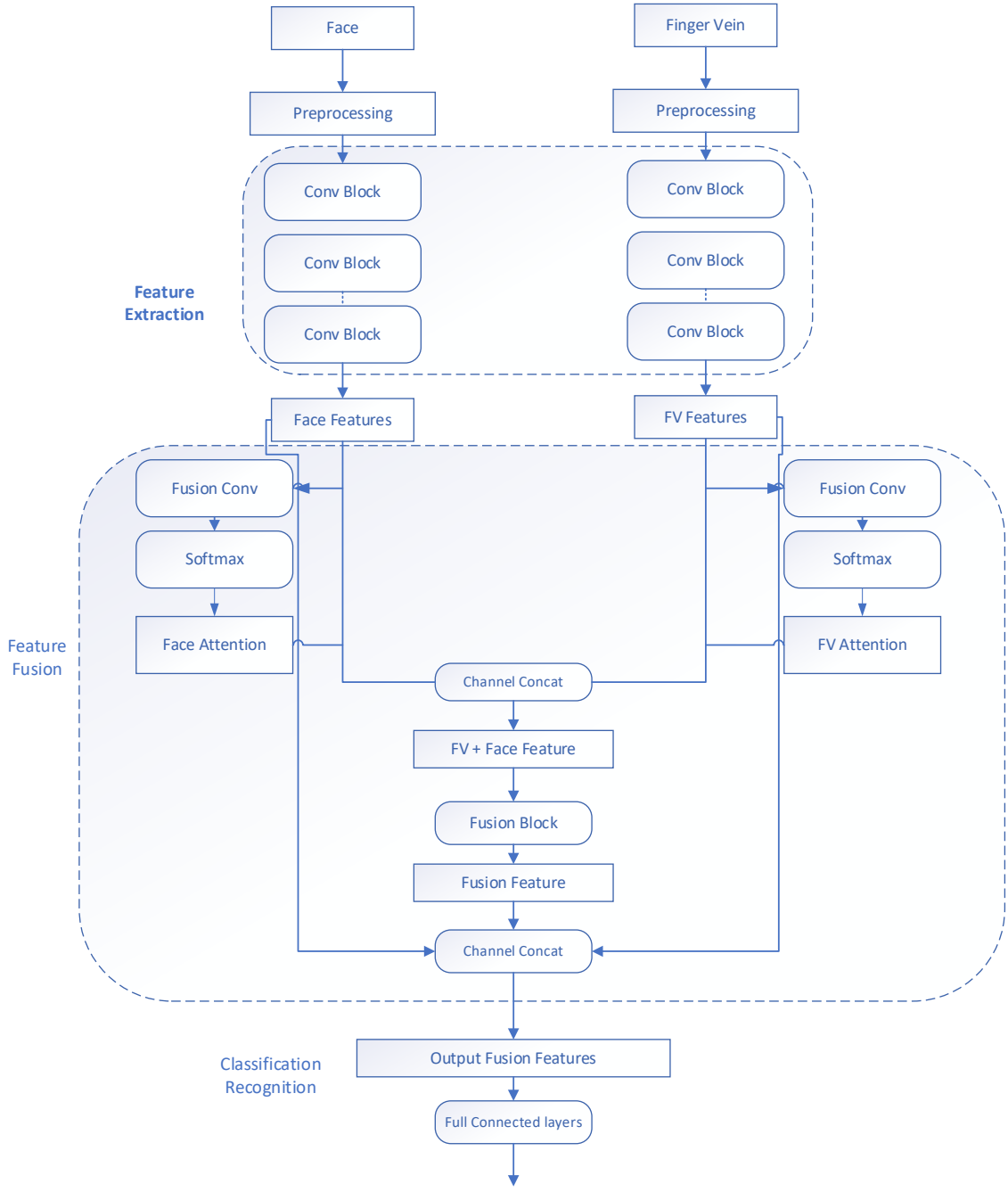

Figure 5: Feature Fusion Framework for Multimodal Biometrics using CNNs [36]

Their methodology is divided into three stages – feature extraction, feature fusion and classification. For feature extraction, the data is preprocessed and then

convoluted into a neural network model, the features are then extracted through the multi-layer convolution and pooling layers. Next is the feature fusion stage where fusion convolution is done to reduce dimensionality and the output is then passed to the softmax layer which obtains the self-attention weights and multiplies the features obtained from the feature extraction and then channel concat combines the two set of features together to get a fusion of the features extracted from finger vein and face. To prevent information loss, the features from the feature extraction step are obtained. The classification step is mainly done in the fully connected layer of the CNN. AlexNet and VGG-19 were used as the networks of choice with their fully connected layers discarded. Only convolutional and pooling layers before the fully connected layers were used.

The results obtained can be seen in the table 17 & 18 showing both the accuracies obtained by using unimodal and multimodal experiments.

Dataset / Model	Parameter Quantity	Test Set Accuracy		
		SDUMLA-FV	USM-FV	CASIA-WebFace
AlexNet-Fusion	16,630,440	0.7020	0.4561	0.5395
VGG-19-Fusiom	143,667,240	0.8757	0.6734	0.5575

Table 17: Results of Single-mode Biometrics [43]

Dataset / Model	Parameter Quantity	Test Set Accuracy	
		SDUMLA-FV + CASIA-WebFace	USM-FV + CASIA-WebFace
AlexNet-Fusion	9,858,994	0.9990	0.9935
VGG-19-Fusiom	45,229,938	0.9998	0.9842

Table 18: Results of Multimodal Biometrics [43]

This study showed a multimodal feature layer fusion method that was based on convolutional neural network. Their method introduced the weight of the self-attention mechanism to update the features in the feature fusion step but also uses the separate feature sets from the different modalities to maximize feature information. The highest accuracy obtained was 99.98% which was gotten when VGG-19 was used for fusing the features obtained from SDUMLA-FV and CASIA-WebFace datasets.

2.9 Chapter Summary

In this chapter, we do an overview of studies done in text stylometry, code stylometry, Feature Fusion and Multimodal biometrics. Our proposed method combines text stylometry, code stylometry and feature fusion and we do an overview of multimodal biometrics to show success of modality on another area of research. The results from Multimodal biometrics show that multimodality yields better results that unimodality. We also discuss different level of feature fusion that have been used in previous studies. Selecting the best feature fusion for multimodality is very important for a multimodal experiment to work.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, we do an overview of the data gathering process, experimental design, and research methodology. As discussed in the previous chapters, the aim of our research is to improve classification accuracy of authorship identification by combining the stylometric features from text and source code (multiple modalities). The research goal is to show that authors can be identified better when these features are combined than when they are used independently. To achieve our goal, we need to have a dataset (or corpus) that would include source code and text written by the same author.

Therefore, our research methodology can be defined as, given a set of authors, aptly denoted as A,

Where,

$$A = \{a_1, a_2, a_3, \ldots, a_n\} \quad (9)$$

a set of written text files, symbolized as files,

$$T = \{\{t_1\}, \{t_2\}, \{t_3\} \ldots, \{t_n\}\} \quad (10)$$

and a set of sets of source code files, represented as

$$S = \{\{s_1\}, \{s_2\}, \{s_3\} \ldots, \{s_n\}\} \quad (11)$$

Where, t_n and s_n are also a set of written text and source code, respectively.

For a more granular view, we break down the structure of t_n and s_n, as illustrated in Equations 12 and 13:

$$t_n = \{t_n1, t_n2, t_n3, \ldots, t_nn\} \quad (12)$$

$$s_n = \{s_n1, s_n2, s_n3, \ldots, s_nn\} \quad (13)$$

With this foundational structure in place, the next step involves the construction of an author's feature set, a concatenation of stylometric features from textual files and source code files. This feature set is defined as:

$$a_n = F(t_n) + F(s_n) \quad (14)$$

Where a symbolizes the author in question, F (t_n) represents a compendium of stylometric features extracted from a textual file written by the author, and F (s_n) encapsulates a collection of stylometric features derived from a block of code

(source code) written by the author. This form of feature fusion is known as early stage or feature level feature fusion.

In this chapter, we outline our research design and methodology.

3.2 Data Gathering (Corpus)

We live in the information age, and this is characterized by the rapid growth in the data that can be collected and made available in electronic media [39]. In this digital landscape, the bounty of data that can be amassed and harnessed is virtually boundless, offering an array of opportunities for researchers and scholars.

To create our corpus, we needed to find multiple authors who have written both texts and source code. Furthermore, for our research we decided to focus of source code written in C/C++ to ensure a level of uniformity across the board. In recent times, the internet has been a major source of gathering data that is useful for research.

We scraped data from three different programming tutorials websites[4][5][6]. We chose these websites because they allow various contributors to offer tutorials in multiple programming languages for users to learn how to code. Each tutorial focuses on a particular problem usually determined by the contributor. Also, each tutorial comprised of an explanation (text) of the programming problem and a

[4] www.tutorialspoint.com
[5] www.medium.com
[6] www.geniuspoint.com

corresponding solution (source code) to the problem. In addition, each tutorial was done by exactly one contributor.

Using a python script, we were able to scrape the websites and extracted the tutorials for C/C++. We only extracted tutorials that included both texts and source codes. We tagged every text and source code extracted with their authors. The corpus was deidentified to protect the identity of the authors. We extracted texts and codes for a combined total of 396 authors covering over 3000 different topics: tutorials.com (92), medium.com (217) and genius.com (87). A subsection of the topics covered can be seen in the appendix. It is worth noting however that even though we extracted works written by 396 authors form these websites, not all the authors had both textual and source code documents. We discuss this in the data cleaning section of this chapter.

To provide a glimpse into the data acquisition process, Figure 6 gives a representative snapshot of a typical webpage, illustrative of the content we scraped.

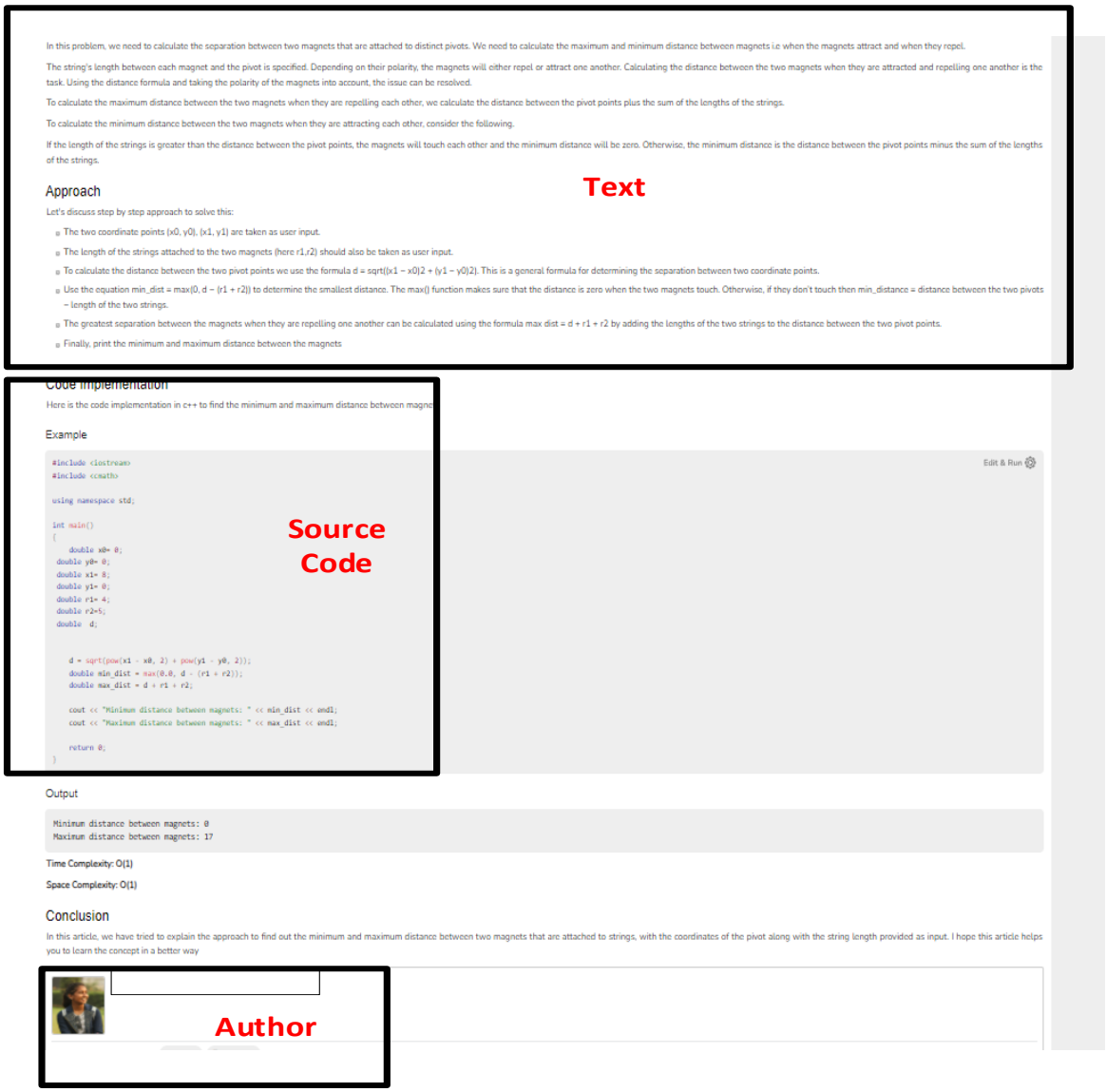

Figure 6: Sample webpage used for data extraction.

3.3 Data Cleaning/Preparation

In any machine learning or stylometry task, an essential part of the process is data cleaning. This is the process of removing unwanted or unnecessary items from the corpus that could cause the results to be inaccurate. However, prior to this cleansing process, a critical criterion was set to lay the foundation for data

quality. We ensured that all selected authors had both text and source code files, eliminating those who possessed only one of these modalities. This thoughtful selection process ensured the coherence and integrity of our dataset. Below we outline the process of cleaning the text documents and the source code documents.

3.3.1 Text Documents

Below, we outline the steps that we undertook to cleanse the text documents.

1. **Stop Words Removal**: Our first stride involved the elimination of stop words. These are commonly occurring words that, while significant in language, tend to be overrepresented and contribute little to the context. Removing them streamlines the dataset, enhancing its focus on more meaningful content.
2. **Removal of URLs**: To safeguard the dataset's coherence, we excised URLs, ensuring that the corpus remained pertinent and devoid of external references. The presence of URLs in textual documents do not provide any information to the identity of the author so every URL was removed from our textual documents.
3. **Removal of Special Characters**: Special characters such as the hyphen (–) or slash (/) are typically deemed as non-contributory and were therefore excluded from our textual documents. The choice to remove specific characters was contingent upon the specific task at hand. For instance, for

research task where currency symbols like "$" have no relevance, they are eliminated.

4. **Removal of Non-English Words**: to maintain linguistic uniformity, non-English words were removed from the corpus.
5. **Exclusion of Documents with Less than 10 Words and 2 Sentences**: A threshold for document length was set, and any document falling short of this criterion was excluded, contributing to data consistency.

Furthermore, term frequency and TFIDF, so additional layers of cleaning were implemented.

1. **Removal of Punctuations**: The elimination of punctuations further refined the text, placing the focus on words
2. **Removal of Single Letters**: Isolated single letters, often devoid of meaningful content, were removed, ensuring that the corpus remained contextually rich.
3. **Converting Digits to Words**: The process included the conversion of digits to their textual representations, preserving the linguistic coherence of the dataset.
4. **Lemmatization**: A linguistic technique, lemmatization, was applied to harmonize different forms of words into their base form. This step ensured that words were consistently represented, irrespective of their inflected forms.

5. **Stemming**: Leveraging stemming, we further condensed words into their root form, harmonizing variations to streamline the dataset.

3.3.2 Source Code Documents

The process of cleansing the source code documents, in comparison to text documents, presented a relatively straightforward task. The specific steps involved in the cleaning process are as follows:

1. **Rectification of Scraping Artifacts**: The initial phase entailed addressing and rectifying scraping artifacts, including but not limited to anomalies like "Ã" and "Â." These artifacts were replaced with spaces to ensure the integrity of the code.
2. **Elimination of Line Padding**: Another integral step in the cleaning process involved the removal of any extraneous line paddings located at the top of the source code.

After the data cleaning and preprocessing stage, we were left with 34 authors with more than 10 textual and source code documents. 50 text files and 50 source code files were randomly selected from each author to best maximize our corpus.. This led us to the selection of a final author list of 19 authors resulting in a total of 950 observations. We selected the number authors and documents because at 19 authors and 50 documents, we can maximize our limited corpus. These parameters made sure that we got the maximum number of observations from our dataset.

3.4 Feature Extraction

This stage of our methodology describes how we extract stylometry features from our corpus (dataset). This is basically the process of converting the text documents and source code documents to their numerical representation so that they can be fed to a machine learning classifier. The stylometric features used can be categorized into four basic categories for text and three categories for source code. All features were extracted using a script written in Python.

These features are identified and defined in tables 19 and 20.

S/N	Category	Features	Explanation
1	**Lexical**	Vocabulary Richness	This can also be called Vocabulary diversity. It is the propensity of an author to avoid the repetition of words. Type Token Ratio was used as a measure of vocabulary richness
		Hapax Legomena	The count of words that appear only once in a document
		Average Word Count	This is the count of words divided by sentence count
		Average Word Length	Average length of words used by the author
		Sentence Count*	Count of sentences
		Term Frequency	This quantifies how frequently a word or term appears in a document
		TFIDF	Also known as Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic designed to quantify the significance of a word within a document.
2	**Character**	Letter Count	Count of all alphabet tokens
		Digit Count	Count of all number tokens
		Punctuation Count	Count of all punctuation tokens

		Upper Case Count	Count of all upper-case tokens
		Lower Case Count	Count of all lower-case tokens
3	**Syntactic**	Verb Count	Count of all verbs in the document
		Noun Count	Count of all words that are nouns.
		Adverb Count	Count of adverb words
		Adjective Count	Count of adjective words
4	**Semantic**	Function words	These are words that signal grammatical relationships but are less central to expressing meaning. This feature captures the propensity of an author to use function words

Table 19: Textual Stylometric Features

S/N	Category	Features	Description
1	**Lexical**	Log (Number of keywords)	Log of the number of keywords in the code (do, else-if, if, else, switch, for, while)
		Log (Number of Ternary)	This is the log of ternary operators
		Log (number of Token)	This is the number of the word tokens
		Log (number of Comments)	This is the log of the number of comments
		Log (number of literals)	This is the log of the number of string, character and numeric literals
		Log (number of digits)	This is the log of the number of digits in the code
		Vocabulary Richness	This is the number of unique keywords used by the author in the code
		Log (number of functions)	This is the log of the number of functions in the code
		Log (number of Macros)	This is the log of the number of preprocessor directives
		Nesting Depth	This is the highest degree to which control statements and loops are nested
		Branching Factor	This is the branching factor of the tree which is formed by converting code blocks into nodes

		Standard Deviation of Number of Parameters	This is the standard deviation of the number of parameters of the functions in the code
		Average Line Length	The average length of each line
		Standard Deviation of Line Length	This is the standard deviation of the character lengths of each line
2	**Layout**	Maximum Line Length	This is the length of the longest line in the code
		Minimum Line Length	This is the length of the shortest line in the code
		Average Indentation	This is the average number of indentations in the code
		Code Block levels	This is the number of levels of code blocks
		Leading Space Count	This is the count of leading spaces in the code
		Tab Characters Count	This is the count of tab characters in the code
		Space Characters count	This is the count of space characters in the code
		Count Empty lines	This is the count of empty lines in the code
		Whitespace ratio	This is the ratio between the number of whitespaces (spaces, tabs and newlines) and non-whitespace characters.
		New Line before open brace	A Boolean representing whether most of the code block braces are preceded by a newline or not
		Tab Leading lines	A Boolean representing whether most indented lines begin with spaces or tabs.

3	**Syntactic**	AST Node Type Term Frequency	Term Frequency of AST node types excluding leaves
		AST Node types TFIDF	TFIDF of node types excluding leaves
		AST Node types of Average Depth	Average depth of AST node types excluding leaves
		C++ Keywords	Term frequency of C++ Keywords
		Code 1n AST Leaves Term frequency	This is the term frequency of code unigrams in AST leaves
		Code 1n AST Leaves TFIDF	This is the TFIDF of the code unigrams in AST Leaves
		Code 1n AST Leaves Average Depth	This is the average depth of the code unigrams in AST leaves
		Max Depth AST node	This is the max depth of ab AST node

Table 20: Source Code Stylometric Features

In addition to feature extraction, we employed an approach to feature engineering, ensuring that the extracted features were not only informative but also attuned to the unique characteristics of each document. This involved the division of features (excluding Term frequency and TFIDF because they are already divided by document size when they are calculated) by the length of sentences in textual documents, subsequently excluding sentence length from the final textual feature set. Similarly, in the case of source code documents, each feature was divided by the total length of the source code, quantified by the number of lines.

These strategic adjustments served a twofold purpose. Firstly, they homogenized the features, rendering them comparable across diverse documents. Secondly, the process offered a level playing field, ensuring that the relative magnitude of features did not overshadow their actual significance. Afte the feature extraction

process, we had two datasets, one for text documents with 3,959 features, source code documents with 38,843 features and each with a total of 950 observations (19 authors with 50 documents each).

3.5 Feature Scaling (Normalization)

At this point in our research, we have two distinct datasets, text dataset and source code dataset. Both datasets are a collection of numerical values that epitomize the features extracted from our corpus. However, the feature extraction process, which comprises of counts and term frequencies, had resulted in the creation of a sparse dataset. A Sparse dataset (matrix) is a dataset that is characterized by a preponderance of zero values. The prevalence of sparse dataset is a recurring theme, that is prominent within machine learning and even in entire subfields of machine learning, such as natural language processing and Stylometry. Table 21 shows the percentage of sparsity encountered in our dataset

	Source Code dataset	**Text dataset**
No of Features	38843	3959
No of Non-zero values	163861	85316
Total matrix size	36900850	3761050
Percentage of Sparsity	99%	97%

Table 21: Summary of Datasets showing sparsity

Dealing with sparse matrices as if they were dense (opposite of a sparse matrix where non-zero values are dominant in the dataset) incurs significant computational overhead and could also lead to tainted classification results. To maximize the performance or efficiency of the machine learning classifiers, it is imperative to employ dedicated representations and operations tailored to the

unique challenges posed by dataset sparsity. This approach is known as Normalization.

Feature scaling or Normalization is a preprocessing step in which each input variable is individually scaled to a standardized range, typically from 0 to 1. This range is for floating-point values which offers the optimal precision, ensuring that the data is well-suited for subsequent analysis and modeling. Normalization is implemented to mitigate potential bias in supervised learning models, preventing them from favoring a particular value range. For example, in a linear regression model, if feature scaling is omitted, certain features might exert a disproportionately significant influence compared to others. This imbalance can detrimentally impact prediction accuracy by unfairly elevating the importance of certain variables over others.

For our research, we use min-max scaling method of normalization. Min-max scaling was employed on all the feature columns in our dataset (text and source code dataset). This approach ensures that data across various feature columns is transformed into a consistent range, enhancing the suitability of the dataset for modeling and analysis. Min – max scaling can be defined in equation 15 below.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (15)$$

Where;

- *Xnorm* is the normalized data,
- *X* is the original data,
- *Xmin* is the minimum value within the dataset, and
- *Xmin* stands for the maximum value found within the dataset.

3.6 Feature Selection

3.6.1 Curse of Dimensionality

This is the process of picking or selecting features from our feature set that would give the best output or results for the model. With our dataset containing 38,843 features for the source code dataset, 3959 features for the text dataset and 950 observations, we are confronted with the challenges of high dimensionality. The curse of dimensionality is a phenomenon that arises in machine learning when dealing with high-dimensional data. As the number of features or dimensions increases, the amount of data needed to generalize accurately grows exponentially. In high-dimensional spaces, the volume of the data space expands rapidly, and data points become increasingly sparse. This sparsity can lead to overfitting, where a model performs well on training data but fails to generalize to unseen data. The curse of dimensionality also affects the efficiency of algorithms, as computations become more resource intensive.

To mitigate the curse of dimensionality, techniques such as feature selection, dimensionality reduction, and regularization are employed. Feature selection involves choosing a subset of relevant features, while dimensionality reduction techniques like Principal Component Analysis (PCA) aim to transform the data into a lower-dimensional space while preserving essential information. Regularization methods penalize overly complex models, helping prevent overfitting.

3.6.2 Feature Selection Methodology

This involves the process of cherry-picking the most pertinent features from our feature set. The objective is to assemble a subset of features that will yield the

most robust and meaningful results for our modeling efforts. An ideal situation is for the number of features to be less than the number of observations.

In this work, we employ the use of ANOVA f-test combined with the Extra Tree classifier and 10 – fold cross validation to select the best features from our textual dataset and the best features from our source code datasets. We also introduce the use of hyperparameters so we can get the best possible set of features that will contribute to a better classification accuracy.

3.6.2.1 ANOVA F-Test

ANOVA, the acronym for "Analysis of Variance," stands as a foundational parametric statistical hypothesis test. Its primary mission is to unveil whether the means derived from two or more data samples—often three or more—originate from a shared distribution or diverge significantly.

An F-statistic, or F-test, is a class of statistical tests that delves into the evaluation of variance ratios, comparing variances like those originating from distinct data samples or the explained and unexplained variances arising from a statistical test—in the case, ANOVA. The ANOVA method is a type of F-statistic referred to here as an ANOVA f-test.

It's crucial to grasp that ANOVA assumes a numeric-variable/categorical-variable dynamic, making it ideal for scenarios where one variable is numeric, and the other is categorical. Such situations often manifest in the form of numerical input

variables and a classification target variable, mirroring the configuration of our datasets (text and source code data).

The results of the ANOVA F-test was used for our feature selection stage, helping us to identify and eliminate features that exhibit independence from the target variable. To achieve this in python we used SelectKBest function of the sklearn.feature_selection module. This function takes the dataset as an input and ranks the features using the ANOVA f-test score form the largest score to the smallest score. This function also takes in a score_function which determines which selection function to use which in this case is the ANOVA f-test (f_classif) and k which is the number of features to be selected after the ranking.

3.6.2.2 Hyper-parameter Tuning

A parameter or model parameter is an internal configuration variable of the model, and its value can be inferred based on various reasons. For instance, an internal configuration variable for the ANOVA f-test would be the number of features to select. An hyperparameter is an external configuration setting that lies beyond the internal structure of the model. Hyperparameters are manually specified or tuned based on prior knowledge, domain expertise, or experimentation. In the case of our feature selection, we wanted to select the best features that were dependent on the output variable (author). To achieve this, we couldn't just pick an input number of features to be selected in the ANOVA F-test, so we needed to automatically decide which number of features best contributes to the outcome.

3.6.2.3 Extra Tree Classifier

Much like Random Forest Classifier, Extra Trees leverages the power of multiple decision trees, and aggregates the results of the decision trees to yield a robust prediction. A notable difference is in their approach to feature splitting. While Random Forests uses a greedy algorithm to determine the optimal feature split at each node of a decision tree, Extra Trees takes a more randomized approach. Specifically, Extra Trees introduces an additional layer of randomness by randomly selecting the threshold values for feature splits.

The computational efficiency of Extra Trees emerges as a significant advantage over Random Forests. The randomization in feature splitting not only adds an element of unpredictability but also renders the process much faster. Unlike Random Forests, which evaluate various splitting points to find the optimal one, Extra Trees expedites the process by randomly choosing thresholds. As a result, Extra Trees demonstrates a notable reduction in computational cost, making it an attractive option for scenarios where efficiency is a critical consideration.

3.6.2.4 K – fold Cross – Validation

Cross-validation is a resampling technique employed to assess the performance of machine learning models with a limited dataset. When you have a machine learning model and a dataset, you need to determine how well your model can generalize to unseen data. The typical approach involves dividing your dataset into a training set and a test set. You train your model on the training data and then evaluate its performance on the test data. However, a single evaluation might not be sufficient, as its possible to get a good result by chance. To ensure a more

robust assessment of a model, you want to evaluate it multiple times. This is where k-fold cross-validation comes into play. The parameter "k" specifies how many groups your dataset should be divided into. For our feature selection, we set "k=10" signifying 10-fold cross-validation and we use the 10-fold cross validation combined with an Extra Tree classifier to evaluate the performance of the selected features. We record the accuracy of every iteration of model and use the features that yielded the highest accuracy. The Algorithm 1 below shows our feature selection process.

Input: Dataset *D,*
Output: Selected_Features *F*

1. Split Dataset into input features *X* and output (Class) *Y*
2. Compute min_lenght = 0.4 * Length of (*D*)
3. Compute max_length = 0.95 * Length of (*D*)
4. Set Selected_length = 0
5. Set Selected_features = list()
6. Set max_accuracy = 0
7. For length in range (min_length to max_length):
    i. Compute (Anova f-test, no. of features to select = length) = *Feature_set*
    ii. Build model using ExtraTreeClassifier and 10-Fold Cross Validation = *model*
    iii. Evaluate accuracy(model) = *a*
    iv. IF *a* ≥ Selected_length:
        a) Selected_length = length
        b) Selected_features = Feature_set
8. Return Selected_Features

Algorithm 1: Feature Selection Methodology

3.7 Stylometric Feature Fusion (SFF)

Feature fusion is a crucial step in multimodal stylometry which involves the integration of features extracted from diverse modalities or sources of data, in our case text and source code. This integration allows for the creation of a unified and enriched feature representation that captures valuable information from each modality, thereby enhancing the overall analysis and decision-making process. By synthesizing this comprehensive feature set, the model gains a holistic perspective, enabling more robust and accurate outcomes. This is particularly valuable in the context of our research in authorship identification where multiple modalities could contribute to improved accuracy, which makes feature fusion a pivotal aspect of this work. In this work, we create a novel feature fusion method called the Stylometric feature fusion. We create an algorithm tailored to the fusion of the stylometric features. We employ the use of early fusion, but we modify it by including a feature selection step before and after the concatenation of the features. This can be seen in Algorithm 2.

3.7.1 Early Fusion

Early fusion operates at the feature level. This works by concatenating the feature vectors from both textual dataset and source code dataset into a single, comprehensive feature vector which is used for classification. While this unified feature vector may be substantial in terms of the number of features it comprises, it has the potential to significantly enhance performance when coupled with appropriate learning techniques. This, however, may lead to longer training and

classification times due to the increased feature dimensionality. Early-stage fusion is defined in equation 16 below.

$$a = F(t) + F(s) \qquad (16)$$

Where a is the author, F(t) is a set of stylometric features extracted from a textual file written by the author and F(s) is a set of stylometric features extracted from a block of code written by the author.

Following the feature selection process for both textual and source code documents, we were left with 587 textual features and 597 source code features with 950 observations. After the feature fusion and the second feature selection stage that we introduced, which only implies to multimodality, we had a third multimodal dataset with 798 number of features and 950 number of observations. Algorithm 2 shows our feature fusion process.

Input: Selected_text_features F(T), Selected_code_features F(C)
Output: Selected_Multimodal_features F(M)

1. Set Multimodal_features *M* = F(T) + F(M) as seen in equation 16
2. Computer F(M) = Algorithm 2(M)
3. Output F(M)

Algorithm 2: Feature Fusion Process

3.8 Machine Learning Classifiers

This phase represents the cognitive nucleus of our research, the hub of knowledge discovery. Within our methodology, we employ multiple classifiers: Random Forest, Naïve Bayes, Multilayer Perceptron (MLP), Extremely Randomized Trees and Support Vector Machines. The selection of these

algorithms is based on their established success in prior authorship identification studies.

3.8.1 Random Forest

Random Forests, often referred to as Random Decision Forests, represent a robust ensemble learning technique employed for a diverse array of tasks, encompassing classification, regression, and more. The essence of this method lies in the creation of a multitude of decision trees during the training phase. In classification tasks, the output of a Random Forest corresponds to the class that most of the decision trees select. Conversely, for regression tasks, the collective prediction from the individual trees is usually the mean or average. Its remarkable prowess is underscored by its consistent delivery of exceptional performance across an extensive spectrum of classification and regression predictive modeling challenges. This algorithm's versatility has solidified its status as a favorite in the field of machine learning [40], [41], [20], [42].

A noteworthy attribute of Random Decision Forests is their ability to mitigate a common pitfall encountered with decision trees—overfitting to the training dataset.

3.8.2 Support Vector Machines

Support Vector Machine (SVM) is a machine learning algorithm renowned for its versatility across a broad spectrum of tasks, including both linear and nonlinear classification, regression, and even the detection of outliers. Its utility extends to a myriad of applications, ranging from text classification [43], image classification [44], spam detection [45], and handwriting identification [46], to gene expression analysis [47], face detection [48], and anomaly detection [49]. SVMs exhibit adaptability and efficiency, making them a preferred choice in scenarios

involving high-dimensional data and intricate nonlinear relationships. The underlying principle that empowers SVM's effectiveness lies in its pursuit of identifying the optimal separating hyperplane, one that maximizes the margin between different classes within the target feature.

The fundamental goal of the SVM algorithm is to identify the optimal hyperplane within an N-dimensional space. This hyperplane's crucial role is to effectively separate data points across different classes residing within the feature space. The optimization objective for this hyperplane is to maximize the margin between the closest points from distinct classes, ensuring the greatest possible separation. The dimensionality of this hyperplane directly correlates with the number of input features. In cases where there are precisely two input features, the hyperplane takes the form of a simple line. With three input features, the hyperplane transforms into a 2-D plane. However, as the number of features exceeds three, visualizing the hyperplane's configuration becomes increasingly complex.

This algorithm, while computationally demanding, yields robust results, making SVM an asset in the field of machine learning.

3.8.3 Naïve Bayes (Gaussian)

Gaussian Naive Bayes is a fundamental machine learning algorithm that falls under the Naive Bayes family. It's particularly popular for classification tasks, especially in the field of text classification and sentiment analysis. The "Gaussian" in its name signifies that it assumes that the features associated with each class follow a Gaussian (normal) distribution.

One of the core principles behind Gaussian Naive Bayes is its application of Bayes' theorem, which enables the model to calculate the probability of a certain class given the observed features. Despite its simplicity, this algorithm has proven to be quite effective in many scenarios [20],[50],[51]. By assuming that features are conditionally independent of each other, it simplifies the model-building process while still yielding reasonable performance.

Gaussian Naive Bayes is widely used in various domains, including spam email detection[52], document classification[53], and medical diagnosis[54]. Its speed and reliability make it a valuable tool for quick, preliminary classification tasks.

3.8.4 Multilayer Perceptron (MLP)

Multilayer perceptron (MLP) is a type of artificial neural network widely used in the field of machine learning and deep learning. It's known for its capability to handle complex problems, especially those involving non-linear relationships between input and output data. MLP is a feedforward neural network consisting of an input layer, one or more hidden layers, and an output layer. Each layer is composed of neurons (or nodes) that perform weighted sum calculations and apply activation functions to produce the output. These layers are interconnected with weighted connections that allow the network to capture intricate patterns in the data.

One of the key strengths of MLP is its capacity to model complex relationships within data, making it suitable for various tasks like regression[55][56], classification[57][58], and pattern recognition [59]. Training an MLP involves using a supervised learning approach, typically backpropagation, where the network

adjusts its weights to minimize the error between predicted and actual outputs. While MLPs are known for their versatility, they often require a substantial amount of labeled training data to perform well.

In recent years, multilayer perceptron has been integrated into larger and more advanced deep learning architectures[60]. By stacking multiple layers of MLPs and incorporating techniques such as dropout and batch normalization, deep neural networks have achieved remarkable success in tasks like image recognition, natural language processing, and reinforcement learning. Despite their capabilities, MLPs also come with challenges, such as overfitting and difficulties in training very deep networks. Researchers continue to innovate, working on novel approaches to address these issues and further enhance the power of multilayer perceptron.

3.9 Evaluation Metrics

Machine learning evaluation metrics are critical tools used to assess the performance of a machine learning model. These metrics provide quantifiable measures of how well a model is doing and are essential for understanding its strengths and weaknesses. Common evaluation metrics vary depending on the type of machine learning task. For classification tasks, metrics like accuracy, precision, recall, F1 score, and the ROC curve are commonly used.

In regression tasks, metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) measure the error between predicted and actual values. These metrics help quantify how close the model's predictions are to the ground truth. Additionally, for probabilistic models, log loss

or cross-entropy is used to measure the quality of predicted probabilities, especially for binary classification problems. The choice of evaluation metrics should align with the specific objectives and characteristics of the problem, as well as the preferences for minimizing false positives or false negatives, depending on the context.

Since our work is a classification problem, we use accuracy, precision, recall, F1 score, and the ROC curve as our metrics to evaluate the performance of our model.

3.9.1 Classification Accuracy

Classification accuracy is a fundamental and easily interpretable evaluation metric used to gauge the performance of a classification model. It calculates the proportion of correctly predicted instances out of the total number of instances in a dataset, providing a simple and intuitive measure of the model's correctness. For instance, if a classifier is tasked with distinguishing between cats and dogs, and it classifies 95 out of 100 images correctly, the accuracy would be 95%, indicating that it is accurate in its predictions for 95% of the cases. This metric is especially helpful when the dataset has a balanced class distribution, where each class has roughly the same number of instances. Given a two-class classification problem, where the model can only classify an instance as positive or negative, classification accuracy can be defined as

$$Classification\ Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad (17)$$

Where:

TP = Number of correctly classified positive instances

TN = Number of correctly classified negative instances

FP = Number of incorrectly classified positive instances

FN = Number of incorrectly classified negative instances

However, classification accuracy may not be the most appropriate metric in all situations. It doesn't account for imbalanced datasets, where one class heavily outweighs the others. In such cases, a model might achieve a high accuracy by predicting the majority class correctly while completely missing the minority class. In these scenarios, other evaluation metrics like precision, recall, F1 score, or the area under the Receiver Operating Characteristic curve (AUC-ROC) are often preferred, as they provide a more comprehensive assessment of the model's performance, especially regarding its ability to identify positive instances and avoid false positives or negatives.

3.9.2 Precision

Precision is a crucial model evaluation metric in machine learning, particularly in classification tasks, where it focuses on the accuracy of positive predictions made by a model. It quantifies the proportion of true positive predictions (correctly identified positive cases) out of all instances classified as positive, whether they are true or false positives. In other words, precision measures the model's ability to make accurate positive predictions without producing a high rate of false alarms. High precision implies that the positive predictions are reliable,

making it a valuable metric in applications where false positives have significant consequences, such as medical diagnoses or fraud detection. Given a two-class classification problem, where the model can only classify an instance as positive or negative, classification accuracy can be defined as

$$Precision = \frac{TP}{TP+FP} \qquad (18)$$

Where:

TP = Number of correctly classified positive instances

FP = Number of incorrectly classified positive instances

A higher precision value indicates that the model makes more cautious and conservative positive predictions, which can be beneficial in scenarios where false positives are costly. However, in applications where missing positive instances (lower recall) is more problematic than an occasional false alarm, it's important to strike the right balance between precision and recall ensuring the model's performance aligns with the specific objectives of the task.

3.9.3 Recall

Recall, often referred to as sensitivity or true positive rate, is a fundamental metric in machine learning that assesses a model's ability to identify all positive instances in a dataset. It measures the proportion of true positive predictions (correctly identified positive cases) out of all actual positive instances. In other words, recall quantifies how effectively a model captures and retrieves relevant data points from the dataset. High recall implies that the model excels at identifying as many positive instances as possible, which is crucial in applications where

missing a positive case could have severe consequences, such as disease diagnosis or security screening. Given a two-class classification problem, where the model can only classify an instance as positive or negative, classification accuracy can be defined as

$$Recall = \frac{TP}{TP+FN} \qquad (19)$$

Where:

TP = Number of correctly classified positive instances

FN = Number of incorrectly classified negative instances

A higher recall value suggests a model that is conservative in making positive predictions, striving to minimize the risk of missing relevant instances. Striking the right balance between precision and recall is essential, as it depends on the specific objectives and requirements of the machine learning task.

3.9.4 F1 Score

The F1 score is a widely used metric in machine learning that strikes a balance between precision and recall. It is particularly valuable in situations where class imbalance is prevalent or when both false positives and false negatives are costly. The F1 score combines these two important aspects of classification performance into a single value, making it a reliable indicator of a model's overall effectiveness. It is calculated by taking the harmonic mean of precision and recall, providing a single score that reflects how well a model correctly classifies instances of the positive class while minimizing both false positives and false negatives. In

essence, the F1 score considers not only the correctness of positive predictions (precision) but also the ability to capture all actual positive instances (recall).

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \qquad (20)$$

The F1 score is especially useful when it is essential to balance precision and recall, and its harmonic mean nature gives higher weight to the lower of the two. This means that the F1 score will be lower if either precision or recall is significantly lower than the other, making it a robust metric for evaluating models in various applications, including medical diagnostics, information retrieval, and fraud detection. The F1 score's value falls between 0 and 1, where higher values indicate better classification performance. It helps with finding a suitable trade-off between precision and recall, depending on the specific requirements and constraints of their classification tasks.

3.9.5 ROC AUC score

The ROC AUC (Receiver Operating Characteristic Area Under the Curve) score is a fundamental performance metric for classification models. The ROC AUC score provides a single value that summarizes the classifier's overall performance, making it easier to compare different models. An AUC score of 0.5 indicates a random classifier, while a score of 1 suggests a perfect model that can separate the two classes. In practice, most classifiers aim for an AUC score greater than 0.5, signifying their effectiveness in making accurate predictions.

The ROC AUC score is particularly useful when dealing with imbalanced datasets, where one class significantly outnumbers the other. It helps assess the model's ability to correctly rank instances and is robust to class distribution, making it a

preferred choice for evaluating classifiers in a wide range of applications, including medical diagnostics, credit risk assessment, and spam email detection. A higher ROC AUC score implies better discrimination power, and it serves as a valuable tool for selecting the most suitable model for a classification task.

We identify that based on the very balanced nature of our dataset (all authors have equal number of documents); classification accuracy is enough to evaluate the effectiveness of our models. However, we introduce the use of the other metrics to give a better understanding of how well our methodology performs.

3.9.6 Workflow of methodology

The workflow commences with the data gathering phase, where data is sourced and collected from multiple websites, creating our source code documents and text documents (corpus). The next stage is the data preprocessing and feature extraction stage which involves cleaning and filtering of the corpus. Following this, feature extraction takes place, focusing on extracting stylometric features from both text and source code documents. Feature fusion which combines the features from source code and text into one representation (Multimodal Features).

The next and final stage is Classification, perhaps the heart of this work, which involves employing robust algorithms to build models that differentiate and identify authors accurately. In essence, the workflow acts as a roadmap guiding the research process from data acquisition to meaningful results.

Figure 7 illustrates the workflow adopted in this research to execute our methodology effectively.

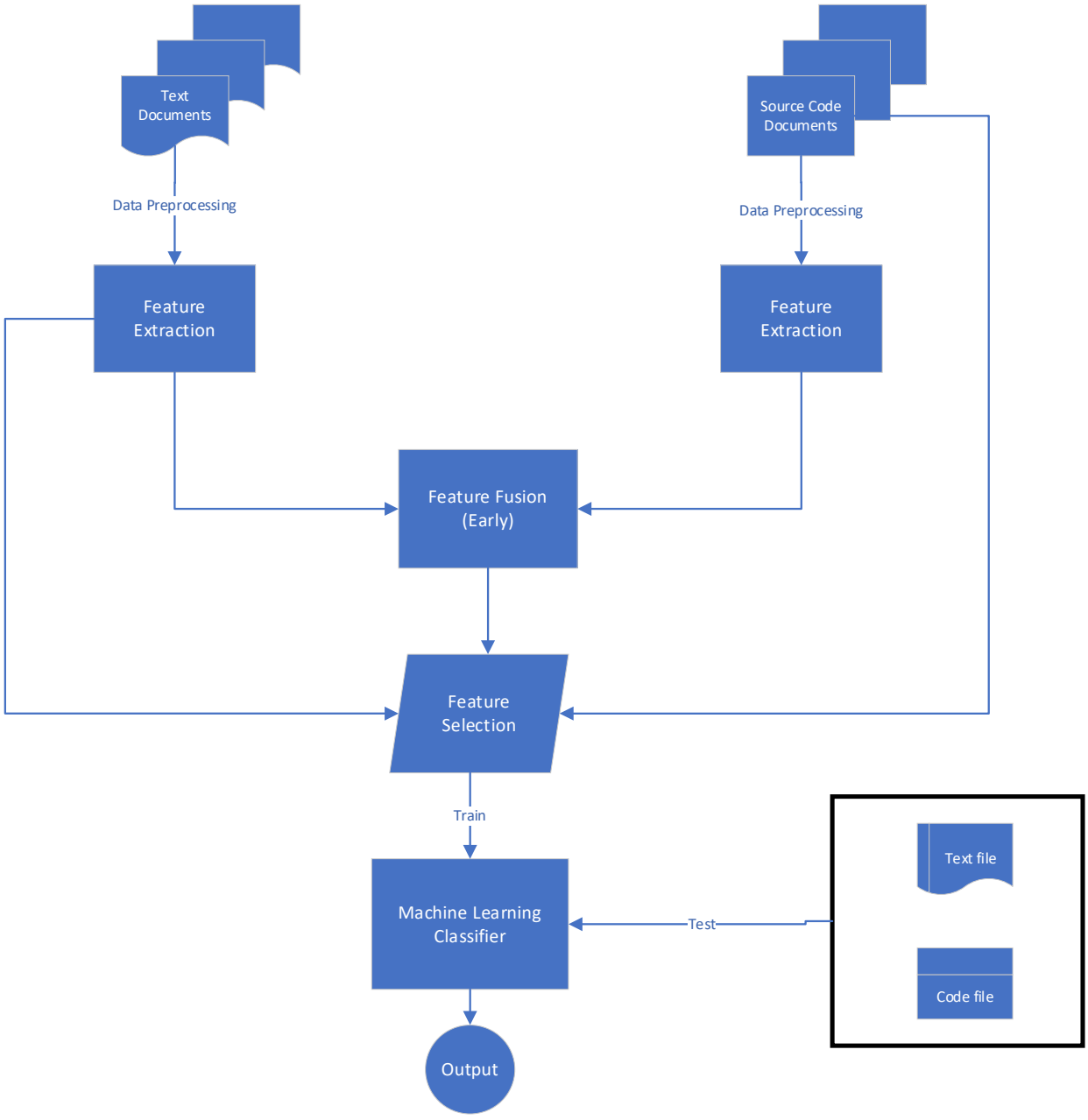

Figure 7: Methodology Workflow

CHAPTER FOUR

RESULT EVALUATION

4.1 Introduction

In this chapter, we discuss and evaluate the results obtained from carrying out authorship identification using multimodal stylometry by comparing the results with single mode stylometry which is our baseline method. We also carry out a second experiment using our methods to distinguish between human generated text and source code from machine generated text and source code (ChatGPT[7]

[7] https://openai.com/

4.2 Experimental setting

Our evaluation focused on the application of multimodal stylometry techniques to our dataset, referred to as MMF. This dataset comprises 19 distinct authors, each contributing 50 documents for both text and source code, resulting in a total of 950 observations. After the feature selection process described in chapter 3, we were left with 597 features for source code data and 587 numbers of features for textual data. The selected features from both modalities are then concatenated to form a multimodal dataset. This dataset also goes through the feature selection process. After feature selection, the multimodal dataset has 798 features, comprising of 458 source code features and 340 textual features.

To comprehensively evaluate the effectiveness of our proposed multimodal approach, we utilized three machine learning algorithms, Gaussian Naïve Bayes, Multilayer perceptron, and Random Forest (Support Vector Machines was excluded from this experiment due to its very poor performance on the dataset) combined with a robust 10-fold cross-validation methodology. This process involved randomly partitioning the dataset into ten subsets, allocating 90% for training and reserving the remaining 10% for testing. To ensure the reliability and consistency of our evaluation, we iterated this 10-fold cross-validation procedure three (3) times. This approach was particularly well-suited to our research because our dataset size was relatively small, and cross-validation becomes crucial in such scenarios to mitigate the limitations associated with limited data availability.

The decision to perform three iterations of the 10-fold cross-validation process was a deliberate choice aimed at enhancing the reliability of our results. By repeating this comprehensive evaluation multiple times, we were better equipped to account for any potential variations that could occur due to the randomization inherent in the cross-validation process. This approach allowed us to extract more robust insights into the performance of our multimodal stylometry techniques under different conditions and settings, thus strengthening the validity of our findings. The eventual evaluation metric (classification accuracy, precision, recall, F1 and ROC AUC) score was determined by calculating the mean of the individual scores obtained from each individual run of the classifier. This method allowed us to consolidate the results from the multiple iterations, providing a more stable and representative measure of the classifier's performance. It effectively reduced the impact of any potential variability that might arise from individual runs and provided a comprehensive assessment of our proposed multimodal stylometry approach.

To establish a baseline for our results, we compared the performance of our multimodal stylometry approach with the results obtained from single mode stylometry (source code and text). The baseline methods involved utilizing 50 documents for both the text and source code datasets within the same set of authors (19), allowing us to gauge the improvements gained from adopting multimodal techniques.

4.2.1 Open World vs Closed World

In machine learning and artificial intelligence, two fundamental paradigms shape the way systems approach data and decision-making: the closed world and open world paradigms. These paradigms encapsulate the varying degrees of knowledge and adaptability that AI systems exhibit, ultimately influencing their behavior in different contexts.

In an open world setting, AI systems acknowledge the vastness and dynamism of the real world. There is an understanding that the training data and the knowledge encapsulated in it are inherently incomplete, and the system is designed to adapt, learn, and make informed decisions even in the presence of novel or unanticipated data. Open world systems can identify unknown entities and respond more flexibly to evolving situations. These systems are pivotal in handling the complexity of real-world data but may face challenges in maintaining boundaries and certainty in situations where closed world assumptions could provide more stable results.

Conversely, in a closed world scenario, the system operates under the assumption that the knowledge it possesses about the world is exhaustive and complete. This implies that the system recognizes and can make decisions only about entities and concepts that are well-defined and explicitly accounted for in its training data. Any input that does not align with this predefined knowledge is regarded as unknown or anomalous, often leading to rejection or incorrect classifications. Closed world systems tend to excel in well-defined, controlled environments but face limitations when confronted with real-world data that is dynamic, diverse, and unstructured.

In this research, we adopt a closed-world approach, focusing on a multiclass classification task, which implies that all possible classes or categories are known and predefined, and the model's objective is to classify data into one of these pre-established categories. This approach was selected because it simplifies the classification task by assuming that data will belong to one of the known classes and this also makes it a practical choice for real-world applications. Throughout this research, we evaluate the performance of our models under the closed-world assumption to gain insights into the effectiveness of our proposed approaches for the multiclass classification problem.

4.3 Result Evaluation

In this section, we show the results obtained from identifying 19 distinct authors based on the analysis of their written text and source code and by combining the features obtained from both modalities (Multimodal Stylometry). Our approach follows a closed-world assumption, which implies that we can only recognize authors who are part of our predefined author set. Text or source code attributed to authors not included in our established set would inevitably be misclassified, possibly as one of the known authors within the set. This concept also holds relevance for addressing issues like ghostwriting. Our dataset encompasses 50 text files and 50 source code files for each of the 19 authors, leading to a comprehensive evaluation of our multimodal feature set. We conducted a comparative analysis of the results yielded by our multimodal

approach against those achieved through single mode stylometry, as illustrated in Table 22 below.

No of Authors: 19 **No of Documents**: 50 **Total Number of Instances**: 950 **Feature Fusion % split**: ***Text*** (42.6%), ***Code*** (57.4%)							
Classifiers	**Modality**	**No of Features**	**CA (%)**	**P (%)**	**R (%)**	**F1 (%)**	**AUC (%)**
Gaussian Naïve Bayes	Code	597	39.6	38.4	39.6	36.4	75.5
	Text	587	41.8	41.6	41.8	39.6	74.7
	Multimodal	798	**48.8**	**48.0**	**48.8**	**46.5**	**77.2**
Random Forest	Code	597	43.3	43.3	43.5	42.1	82.8
	Text	587	55.4	56.9	55.8	54.9	91.0
	Multimodal	798	**58.4**	**58.1**	**58.1**	**57.3**	**91.2**
MLP	Code	597	43.4	46.6	43.2	43.8	86.3
	Text	587	52.0	55.2	52.2	52.6	90.0
	Multimodal	798	**56.3**	**58.5**	**56.4**	**56.1**	**90.8**

Table 22: Results obtained from unimodal and multimodal stylometry.

Where;

CA: Classification Accuracy

P: Precision

R: Recall

F1: F1 Score

AUC: AUC ROC

The comprehensive examination of the results depicted in the table above shows the notable superiority of multimodal stylometry in contrast to single mode stylometry across all our evaluation metrics. This advantageous performance

manifests in terms of precision, recall, F1 score, and accuracy. The noteworthy exception lies in the AUC ROC score, multimodality slightly outperforms text stylometry by a meagre 0.001 margin. Intriguingly, it is paramount to acknowledge that the Random Forest classifier exhibits a superior overall performance compared to other classifiers employed in the study in terms of the classification accuracy.

The discernible improvement brought about by multimodal stylometry becomes particularly apparent when analyzing the post-fusion outcome of both code and text stylometric features. Following the fusion process, our feature selection algorithm identified and retained 798 salient features. This feature selection significantly contributed to the refined performance of our models. The discriminative power inherent in these selected features accentuates the efficacy of a multimodal approach, demonstrating its capacity to leverage complementary information from both text and source code domains.

4.3.1 Precision vs Recall

As highlighted in the previous chapter, precision and recall are two crucial metrics in evaluating the performance of classification models. Precision measures the accuracy of positive predictions by assessing the proportion of true positives among all instances predicted as positive. In other words, precision gauges the model's ability to avoid false positives. On the other hand, recall, also known as sensitivity or true positive rate, evaluates the model's capability to capture and correctly identify all relevant instances in the dataset. It calculates the proportion of true positives among all actual positive instances. Precision and recall are often

in tension; as one increases, the other may decrease. Striking the right balance between precision and recall is essential, depending on the specific goals and requirements of a given application. Achieving high precision is crucial when minimizing false positives is a priority, while emphasizing recall becomes imperative when ensuring the comprehensive identification of positive instances is paramount.

In the presented results table, it is evident that the utilized classifiers generally maintain moderately balanced recall and precision scores. Notably, both the Random Forest and MLP classifiers achieve scores exceeding 50%, indicating their proficiency in minimizing false positives (precision) and accurately identifying all pertinent instances in the dataset (recall). However, it is crucial to clarify that the primary objective of this study was not to ascertain whether the classifiers could attain a higher classification accuracy or recall individually. Instead, the focus was on demonstrating that multimodal stylometry outperforms single mode stylometry, considering both text and source code. The results unequivocally indicate that multimodal stylometry excels in minimizing false positives (precision) and capturing all relevant instances in the dataset (recall), surpassing the performance of utilizing features exclusively from either text or code stylometry.

4.4 Dataset Scalability

Additionally, our investigation delved into exploring the impact of the number of documents per author on the task of authorship identification. Following the acquisition of the initial dataset, we implemented a controlled experiment by

randomly selecting a subset of N documents for authors with document counts greater than or equal to N. This strategic approach allowed us to systematically vary the size of the dataset, shedding light on how the quantity of documents influences the efficacy of the authorship identification process and Multimodal Stylometry. Such an inquiry into the dataset's scalability holds practical implications, offering insights into the robustness and adaptability of the proposed multimodal stylometry approach under different number of documents, number of authors and dataset size.

Scaling up and down the dataset size is a pivotal aspect of understanding the generalizability and applicability of the proposed methodology. By systematically adjusting the number of documents per author, we gained valuable insights into our methodology’s performance across varying data densities. The objective was to discern whether the multimodal stylometry approach exhibits consistent proficiency in authorship identification across datasets of different sizes. This exploration of dataset scalability provides a more comprehensive understanding of our method's reliability and effectiveness, contributing to the robustness of our findings. The pseudocode below shows our selection process. It is worth noting that we increase the number of documents, the number of authors is reduced and vice versa. All the datasets went through the same features selection and feature fusion processes.

Input: Corpus *C,*
Output: New_Corpus where X = N *N_C*

1. X = number of documents per author
2. N = number of documents to be selected
3. If X >= N:
    i. Select author.
    ii. Random (Select N(documents) where N(documents) is a subset of X(documents))
4. Else:
    i. Discard author

Algorithm 3: Dataset Scaling

Algorithm 3 is executed for N = (10, 15, 20, 25, 30, 35, 40, 45, 55, 60, 65, 70) and gives an additional 12 datasets (7 datasets with document size smaller than our original dataset and 5 with documents size larger than our original dataset). The sections that follow show the results obtained from experimentation on all 12 additional datasets. Also, we can see the effect of an increase or decrease in the number of documents.

4.4.1 Dataset One

No of Authors: 34
No of Documents: 10
Total Number of Instances: 340
Feature Fusion % split: ***Text*** (48.9%), ***Code*** (51.3%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	232	25.4	19.8	25.4	21.4	74.2
	Text	266	42.1	31.6	42.1	34.6	73.7
	Multimodal	269	**43.5**	**36.1**	**43.5**	**38.4**	**77.4**
Random Forest	Source Code	232	36.9	31.2	37.2	31.3	79.3
	Text	266	53.4	49.8	54.5	51.0	89.5
	Multimodal	269	**57.2**	**51.1**	**56.5**	**52.8**	**90.0**
MLP	Source Code	232	35.1	28.6	35.3	30.5	81.2
	Text	266	**55.6**	**50.5**	**54.6**	**51.6**	90.8
	Multimodal	269	54.4	49.1	54.1	50.8	**91.0**

Table 23: Results obtained for unimodal and multimodal stylometry using 10 documents.

Table 23 represents the results obtained using 10 documents and 24 authors. This gives a dataset with 340 observations. The results show that multimodality outperforms single modality (NB: ≈ 1.4%, RF: ≈ 4%) except with MLP classifier, where text stylometry does better than multimodal stylometry. This result shows that multimodal stylometry doesn't do well on small document size when the MLP classifier is used. Also, we see that though the number of features selected for source code are slightly less than those selected for text, more source code features (51.4%) are selected after feature fusion. This highlights the beauty of multimodality where the best features are selected from each modality to improve model performance.

4.4.2 Dataset Two

No of Authors: 29
No of Documents: 15
Total Number of Instances: 435
Feature Fusion % split: ***Text*** (49.4%), ***Code*** (50.6%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	373	35.7	33.2	35.7	32.0	77.6
	Text	181	38.9	33.8	38.9	34.3	74.9
	Multimodal	344	**43.7**	**38.7**	**43.7**	**39.0**	**75.5**
Random Forest	Source Code	373	42.1	37.8	41.5	37.8	82.2
	Text	181	54.8	53.2	55.8	52.5	90.0
	Multimodal	344	**58.2**	**56.9**	**58.5**	**57.1**	**91.2**
MLP	Source Code	373	40.6	39.7	40.4	37.2	85.8
	Text	181	51.6	51.2	51.9	50.7	90.5
	Multimodal	344	**56.6**	**56.6**	**57.6**	**55.5**	**92.1**

Table 24: Results obtained for unimodal and multimodal stylometry using 15 documents.

Table 24 shows the results obtained when the document size was increased to 15. We see an increase in the performance of multimodal stylometry in comparison

with single mode stylometry. We also see that compared with the results in table 23, MLP performs better with modality with the increase in document size.

4.4.3 Dataset Three

No of Authors: 28
No of Documents: 20
Total Number of Instances: 560
Feature Fusion % split: ***Text*** (51.5%), ***Code*** (48.5%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	511	35.5	35.9	35.5	33.5	76.8
	Text	486	44.5	42.0	44.5	40.1	73.1
	Multimodal	336	**46.3**	**42.4**	**46.3**	**42.6**	**77.6**
Random Forest	Source Code	511	40.5	36.7	39.9	37.1	82.7
	Text	486	55.8	55.8	54.8	54.3	90.3
	Multimodal	336	**57.2**	**55.7**	**56.4**	**55.5**	**91.9**
MLP	Source Code	511	40.8	41.2	40.9	38.7	86.3
	Text	486	53.9	57.3	53.8	53.3	90.6
	Multimodal	**336**	**56.7**	**58.0**	**56.8**	**56.4**	**92.1**

Table 25: Results obtained for unimodal and multimodal stylometry using 20 documents.

As seen in table 25, we see a slight increase in modal performance as document size increased from 15 to 20 though there is a slight drop in the number of authors. We can deduce that document size increase may have more effect on model performance than the number of authors.

4.4.4 Dataset Four

No of Authors: 25
No of Documents: 25
Total Number of Instances: 625
Feature Fusion % split: ***Text*** (43.6%), ***Code*** (56.4%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	279	32.4	31.8	32.4	28.4	74.9
	Text	303	41.5	39.9	41.5	38.4	76.2
	Multimodal	417	**44.4**	**43.5**	**44.4**	**41.5**	**77.1**
Random Forest	Source Code	279	40.0	39.4	40.2	37.8	82.1
	Text	303	53.2	53.1	52.7	51.9	90.3
	Multimodal	417	**56.6**	**56.4**	**56.9**	**55.2**	**91.4**
MLP	Source Code	279	38.6	37.8	38.6	36.4	84.0
	Text	303	48.8	51.5	48.6	48.8	90.1
	Multimodal	417	**53.9**	**55.0**	**53.4**	**53.4**	**91.2**

Table 26: Results obtained for unimodal and multimodal stylometry using 25 documents.

The results shown in table 26 above, we observe that it follows the trends of the previously examined results where multimodality performs better than single mode stylometry. However, there is a slight deep in the overall performance of the classification algorithms.

4.4.5 Dataset Five

No of Authors: 24
No of Documents: 30
Total Number of Instances: 720
Feature Fusion % split: ***Text*** (44.6%), ***Code*** (55.4%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	457	36.2	36.5	36.2	33.7	74.4
	Text	485	38.8	37.5	38.8	35.4	72.7
	Multimodal	628	**42.8**	**41.8**	**42.8**	**40.3**	**74.7**
Random Forest	Source Code	457	37.9	35.1	38.1	35.0	80.6
	Text	485	49.4	48.9	48.8	47.1	88.3
	Multimodal	628	**52.5**	**50.6**	**52.5**	**50.5**	**89.9**
MLP	Source Code	457	42.1	44.0	41.9	40.7	85.5
	Text	485	49.3	51.5	49.5	48.5	89.1
	Multimodal	628	**51.8**	**53.6**	**51.5**	**51.9**	**90.5**

Table 27: Results obtained for unimodal and multimodal stylometry using 30 documents.

The results obtained from dataset five show again that multimodality does a better job at author identification than single modality.

4.4.6 Dataset Six

No of Authors: 21
No of Documents: 35
Total Number of Instances: 735
Feature Fusion % split: ***Text*** (51.4%), ***Code*** (48.6%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	618	39.6	39.8	39.6	37.4	74.3
	Text	494	41.8	41.8	41.8	38.5	73.5
	Multimodal	533	**48.0**	**47.9**	**48.0**	**45.8**	**77.7**
Random Forest	Source Code	618	42.9	41.9	42.5	41.3	83.5
	Text	494	56.3	56.4	56.4	54.8	90.7
	Multimodal	533	**58.6**	**56.7**	**58.3**	**56.0**	**91.3**
MLP	Source Code	618	42.2	45.7	42.6	41.3	86.5
	Text	494	53.0	55.3	53.2	52.5	90.7
	Multimodal	533	**56.7**	**58.6**	**57.1**	**55.9**	**91.9**

Table 28: Results obtained for unimodal and multimodal stylometry using 35 documents.

Table 28 above show the results obtained from increasing the number of documents per author to 35. We begin to see a steady rise in model performance but what is evident is that the gap begins to widen between multimodality classification accuracy and single modality. Also, we see that the feature selection process yielded a reduced number of features for multimodality compared to source code features. This is because the feature selection process is designed to select on features from both modalities that best contributes to the output. There is also a slight drop in the overall accuracy for Naïve Bayes especially for source code stylometry. This is because the curse of dimensionality has little effect on Random Forest and MLP classifiers but can impact the results obtained using Naïve Bayes especially because the optimal number of features selected seemed to be large.

4.4.7 Dataset Seven

No of Authors: 20
No of Documents: 40
Total Number of Instances: 800
Feature Fusion % split: ***Text*** (48.4%), ***Code*** (51.6%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	725	39.8	40.8	39.8	37.9	74.8
	Text	668	42.8	42.8	42.8	39.7	72.9
	Multimodal	641	**44.7**	**45.6**	**44.7**	**42.6**	**75.9**
Random Forest	Source Code	725	45.3	45.1	44.9	42.8	84.5
	Text	668	56.5	57.7	56.7	56.2	91.6
	Multimodal	641	**60.1**	**59.4**	**60.2**	**58.0**	**92.5**
MLP	Source Code	725	41.8	46.5	41.9	41.4	86.4
	Text	668	56.0	59.7	57.0	55.9	90.8
	Multimodal	641	**56.3**	**59.5**	**56.2**	**56.3**	**91.3**

Table 29: Results obtained for unimodal and multimodal stylometry using 40 documents.

We begin to see a slight improvement in model performance across all modalities, but we also see the effect of dimensionality influencing source code stylometry. But the outperformance of multimodality remains constant.

4.4.8 Dataset Eight

No of Authors: 19
No of Documents: 45
Total Number of Instances: 855
Feature Fusion % split: ***Text*** (47.3%), ***Code*** (52.7%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	798	41.1	41.7	41.1	38.8	72.7
	Text	465	40.4	40.8	40.4	38.5	74.0
	Multimodal	692	**47.2**	**48.1**	**47.2**	**45.4**	**75.6**
Random Forest	Source Code	798	44.4	44.8	44.3	40.9	83.2
	Text	465	55.4	56.2	55.4	53.9	90.1
	Multimodal	692	**58.1**	**58.2**	**58.2**	**57.7**	**91.1**
MLP	Source Code	798	46.4	49.4	46.3	46.2	86.8
	Text	465	52.6	55.2	52.1	52.0	89.4
	Multimodal	692	**55.8**	**57.9**	**56.0**	**56.0**	**90.7**

Table 30: Results obtained for unimodal and multimodal stylometry using 45 documents.

The results from dataset 8 is particularly interesting as it maintains the number of authors from our initial experiments, but the number of documents is set to 45. The results show that the difference between multimodality and text modality (which outperforms source code modality) remains the same as our main experiment. We begin to see that author size may not have an outcome on how multimodality would perform against single modality, but it has an influence on the overall performance of the classifiers.

4.4.9 Dataset Nine

No of Authors: 17 **No of Documents**: 55 **Total Number of Instances**: 935 **Feature Fusion % split**: ***Text*** (42.9%), ***Code*** (57.1%)							
Classifier	**Modality**	**No of Features**	**CA (%)**	**P (%)**	**R (%)**	**F1 (%)**	**AUC (%)**
Naïve Bayes	Source Code	684	42.0	44.6	42.0	39.9	76.6
	Text	458	42.4	43.6	42.4	40.7	76.8
	Multimodal	**872**	**51.1**	**51.4**	**51.1**	**49.5**	**78.1**
Random Forest	Source Code	684	43.8	43.0	43.5	41.4	84.2
	Text	458	56.8	58.5	57.3	56.8	91.2
	Multimodal	**872**	**59.8**	**59.8**	**59.3**	**57.5**	**92.2**
MLP	Source Code	684	46.5	49.6	46.7	46.6	87.2
	Text	458	54.4	57.1	54.3	54.4	90.2
	Multimodal	**872**	**58.3**	**60.7**	**58.0**	**58.5**	**92.2**

Table 31: Results obtained for unimodal and multimodal stylometry using 55 documents.

The results in table 31, again show that multimodality outperforms single mode. But we also see that as document size increases multimodality for Naïve Bayes and is not hampered by the curse of dimensionality. This again show the effectiveness of multimodality where the best features from multiple modalities lead to better classification accuracy.

4.4.10 Dataset Ten

No of Authors: 13 **No of Documents**: 60 **Total Number of Instances**: 780 **Feature Fusion % split**: ***Text*** (49.5%), ***Code*** (50.5%)							
Classifier	**Modality**	**No of Features**	**CA (%)**	**P (%)**	**R (%)**	**F1 (%)**	**AUC (%)**
Naïve Bayes	Source Code	532	49.7	55.8	49.7	49.2	80.6
	Text	555	52.7	54.3	52.7	51.4	79.4
	Multimodal	662	**64.5**	**65.7**	**64.5**	**63.6**	**83.8**
Random Forest	Source Code	532	57.9	57.3	57.0	55.5	89.1
	Text	555	71.2	72.3	69.8	69.8	95.4
	Multimodal	662	**72.2**	**73.6**	**72.0**	**71.6**	**95.6**
MLP	Source Code	532	57.6	61.2	58.3	57.6	90.2
	Text	555	66.8	69.5	67.2	67.1	94.4
	Multimodal	662	**72.9**	**75.0**	**73.0**	**72.9**	**95.2**

Table 32: Results obtained for unimodal and multimodal stylometry using 60 documents.

The results in table 32 show that as document size increases, Naïve Bayes and MLP performance for Multimodality increases while the performance for Random Forest classifier decreases. Though we see that Multimodality outperforms single modality, the difference is gradually decreasing when Random Forest classifier is used.

4.4.11 Dataset Eleven

No of Authors: 11
No of Documents: 65
Total Number of Instances: 715
Feature Fusion % split: ***Text*** (50.4%), ***Code*** (49.6%)

Classifier	Modality	No of Features	CA (%)	P (%)	R (%)	F1 (%)	AUC (%)
Naïve Bayes	Source Code	446	48.5	52.0	48.5	45.3	82.5
	Text	436	56.7	59.4	56.7	55.9	82.7
	Multimodal	557	**66.0**	**67.4**	**66.0**	**64.8**	**86.5**
Random Forest	Source Code	446	58.6	59.5	59.0	58.4	91.0
	Text	436	76.3	78.0	75.4	75.2	96.6
	Multimodal	557	**75.9**	**78.1**	**77.0**	**75.0**	**96.7**
MLP	Source Code	446	60.2	62.8	60.5	59.9	92.1
	Text	436	73.7	75.6	74.0	74.0	95.9
	Multimodal	557	**77.4**	**79.5**	**77.1**	**77.1**	**96.4**

Table 33: Results obtained for unimodal and multimodal stylometry using 65 documents.

The results presented in table 33 evidently show that as document size increases, the performance of random forest for modality is decreased compared to single mode but is increases for Naive Bayes and MLP compared to single mode. Model performance, however, is increased across all classifiers.

4.4.12 Dataset Twelve

No of Authors: 10
No of Documents: 70
Total Number of Instances: 700
Feature Fusion % split: ***Text*** (45.5%), ***Code*** (54.5%)

Classifier	**Modality**	**No of Features**	**CA (%)**	**P (%)**	**R (%)**	**F1 (%)**	**AUC (%)**
Naïve Bayes	Source Code	619	56.4	59.1	56.4	55.1	82.8
	Text	392	60.1	61.5	60.1	58.5	85.7
	Multimodal	627	**67.9**	**67.9**	**67.9**	**66.3**	**87.0**
Random Forest	Source Code	619	63.8	63.7	64.3	61.9	92.7
	Text	392	80.0	81.4	80.7	79.5	97.5
	Multimodal	627	**80.7**	**81.9**	**81.6**	**80.3**	**97.8**
MLP	Source Code	619	69.8	72.0	69.6	69.2	94.4
	Text	392	77.7	79.4	77.3	77.3	96.6
	Multimodal	627	**81.9**	**82.8**	**81.7**	**81.7**	**97.3**

Table 34: Results obtained for unimodal and multimodal stylometry using 70 documents.

Finally, we present the results for our dataset with the largest number of documents per author. We see from the results that multimodality outperforms single modality when Naïve Bayes and MLP are used but Random Forest shows little to know improvement with multimodality compared to single modality.

The figures below show a graphical view of classification accuracy as the number of documents increase using our classifiers.

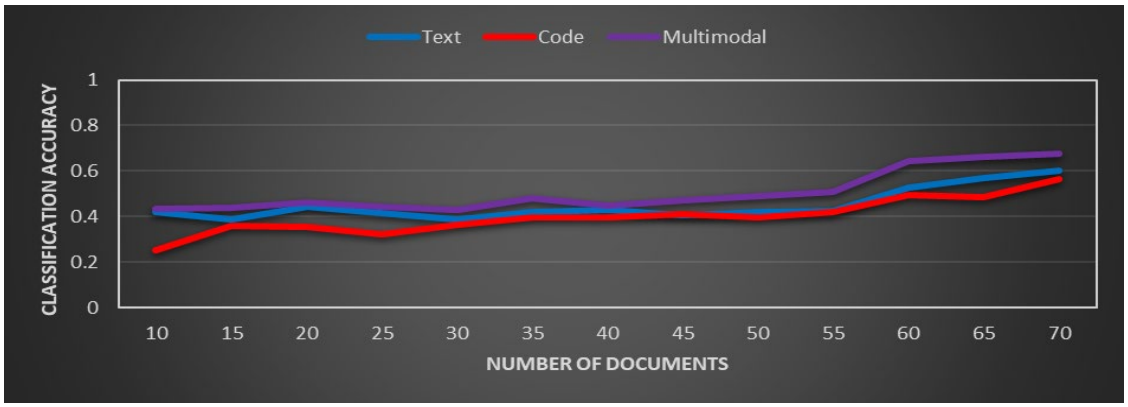

Figure 8: Line Chart showing the effect of scalability as document size increases for unimodal stylometry and multimodal stylometry using Naïve Bayes classifier.

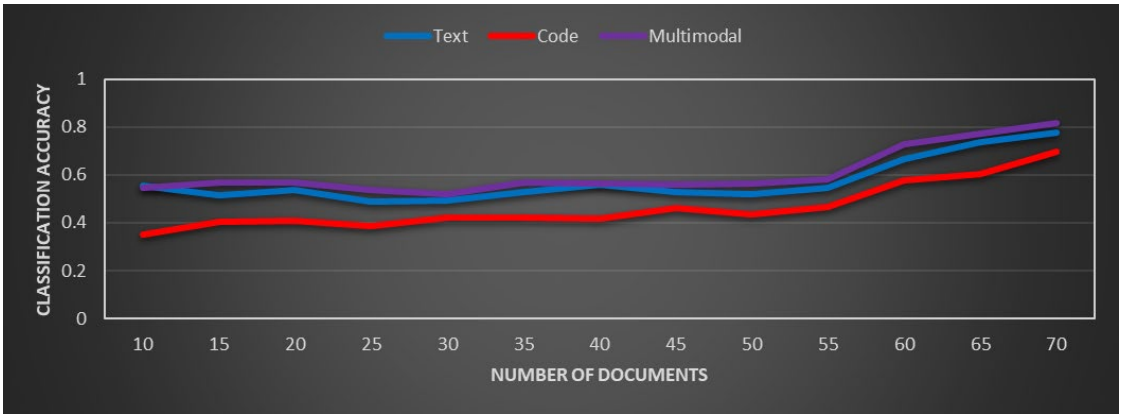

Figure 9: Line Chart showing the effect of scalability as document size increases for unimodal stylometry and multimodal stylometry using MLP classifier.

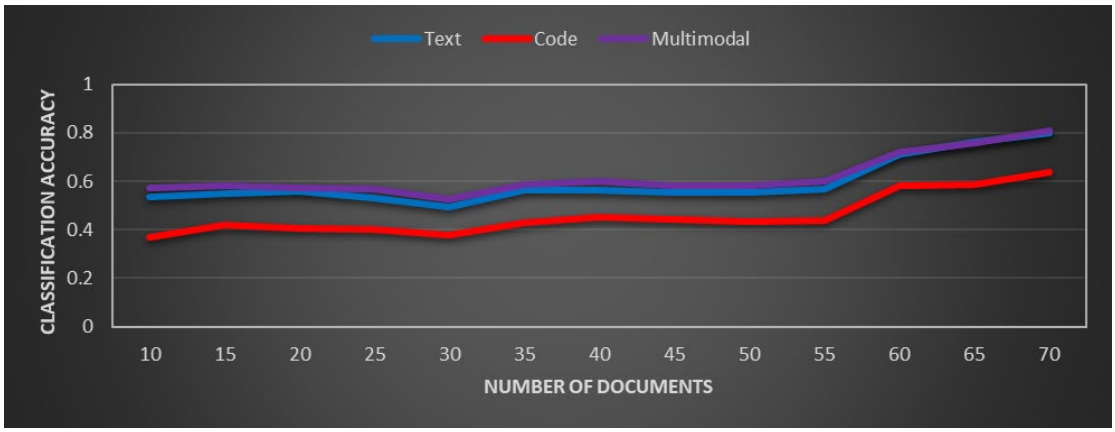

Figure 10: Line Chart showing the effect of scalability as document size increases for unimodal stylometry and multimodal stylometry using Random Forest classifier.

The figures 8, 9 & 10 above show that Naïve Bayes and Random Forest perform well with modality when document size is small. We also see a steady increase in classification accuracy as number of documents increase. As the number of documents begins to exceed 55, we see a see that random forest begins performing poorly for modality but there's increased performance with Naïve Bayes and MLP.

4.5 Distinguish between human and machine generated content.

Distinguishing between human and machine-generated content is a field of study that is increasingly growing due to the rise of sophisticated generative AI models like Chat GPT. Although this models can potentially revolutionize the society, they pose different new challenges. They can be used to produce false news and misinformation. Addressing this challenge aligns with the evolving landscape of language models and the need for discerning between content produced by artificial intelligence (AI) systems and that crafted by human authors.

In the pursuit of this objective, we introduce a research endeavor aimed at distinguishing between human-generated and AI-generated content, encompassing both text and source code. Existing literature has explored manual linguistic distinctions between AI and human writing [61], while Islam et al. [62] have proposed machine learning-based methods for such discrimination using textual content. The results of this exploration will contribute valuable insights into the evolving landscape of AI-generated content and its distinguishability from human-authored material.

In this research, ChatGPT 3.5 and ChatGPT 4.0 serve as our primary sources for machine-generated content. It is noteworthy that OpenAI, the architects of ChatGPT, had previously undertaken efforts to distinguish between human and machine-generated text. However, these endeavors were eventually abandoned due to suboptimal results, with the highest accuracy achieved being a mere 26%, falling below the efficacy of random selection [8]. It is important to contextualize that OpenAI's earlier work aimed at distinguishing human text from a combination of a myriad of AI text generator providers, whereas our research focuses exclusively on content generated by ChatGPT using only Chat GPT 3.5 and Chat GPT 4.0. This distinction underscores the specificity and scope of our investigation, providing a tailored perspective on the capabilities of stylometry to distinguish between ChatGPT-generated content and human generated content.

4.5.1 Experimental Settings

The experiment involved the extraction of 140 samples each of source code and text generated from ChatGPT 3.5 and Chat GPT 4.5, leveraging the OpenAI API. This extraction process was facilitated by presenting the programming topics we scraped from the programming tutorial websites that we used to build our human corpus as questions to the API. We do this to ensure uniformity across human and AI generated content, The ensuing outputs underwent a meticulous cleaning process, aligning with the procedures delineated in Chapter 3, ensuring consistency and comparability with the existing dataset.

[8] https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

In tandem, to enrich the experimental context, a set of 5 text and source code documents were randomly selected from the established dataset from 28 human authors. This selection served to create a diverse and representative set of human-authored content for comparative analysis. We followed the same feature extraction, feature selection and feature fusion processes that were outlined in Chapter 3. It is worth nothing that the text and source code documents extracted from ChatGPT 3.5 and 4 were extracted using topics extracted from our human corpus to ensure uniformity of topics and context of the corpus. After these processes were completed, we had three datasets for source code, text and multimodal. Each dataset had 420 observations. The results table show the number of features that was selected for each of the modalities.

The classification task, differentiating between content generated by humans and ChatGPT, employed the use random forest, Gaussian Naïve Bayes, SVM and Multilayer perceptron classifiers coupled with a 10-fold cross-validation strategy with 3 repeats. This methodological choice was made to ensure comprehensive evaluation and robustness in handling the classification task.

As we can see from the table 35, we see that multimodality outperforms unimodal stylometry across all the classifiers that we use, across all our metrics. Figures 11, 12, and 13 show word clouds for text documents from the human, Chat GPT 3.5 and Chat GPT 4.0 datasets.

No of Authors: 3 (Human, Chat GPT 3.5, Chat GPT 4.5) **No of Documents**: 140 **Total Number of Instances**: 340 **Feature Fusion % split**: ***Text*** (%), ***Code*** (%)							
Classifier	**Modality**	**No of Features**	**CA (%)**	**P (%)**	**R (%)**	**F1 (%)**	**AUC (%)**
Gaussian Naïve Bayes	Code	274	76.0	77.0	76.0	74.7	87.3
	Text	328	71.1	72.3	71.1	69.5	84.4
	Multimodality	383	**79.0**	**80.5**	**79.0**	**78.5**	**88.7**
Random Forest	Code	274	84.8	84.6	84.5	83.9	95.5
	Text	328	80.7	80.0	80.2	79.9	92.8
	Multimodality	383	**88.4**	**89.4**	**89.1**	**88.7**	**96.9**
SVM	Code	274	74.0	75.7	74.0	72.8	86.9
	Text	328	75.7	80.0	75.7	76.0	84.3
	Multimodality	383	**81.1**	**83.6**	**81.1**	**80.2**	**92.0**
MLP	Code	274	83.7	84.9	83.8	83.6	94.0
	Text	328	82.5	84.0	83.2	82.4	93.7
	Multimodality	383	**88.2**	**88.8**	**88.6**	**88.2**	**95.6**

Table 35 : Results obtained for distinguishing between AI and Human generated content using unimodal stylometry and multimodal stylometry.

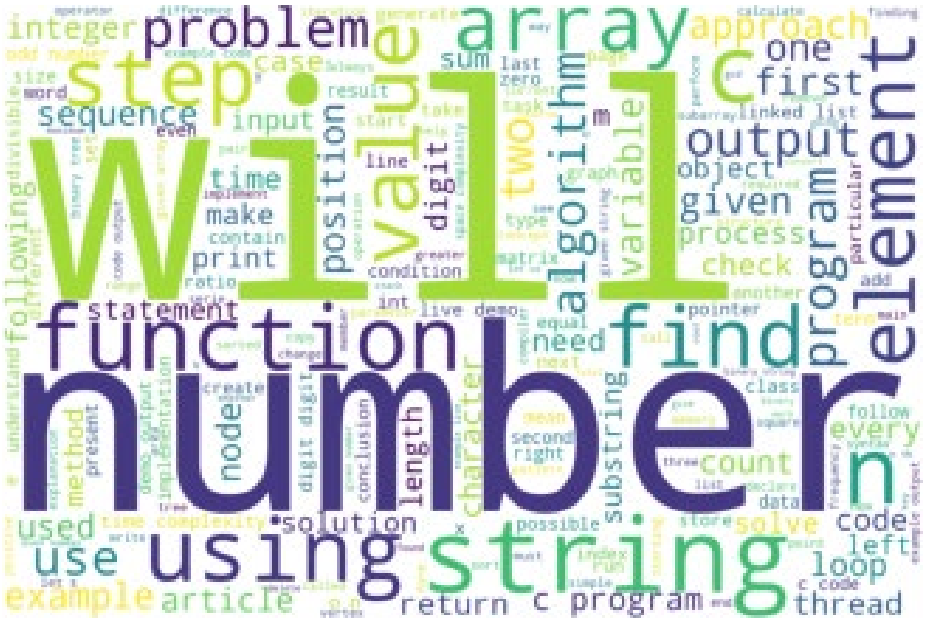

Figure 11: Word Cloud for Human generated text

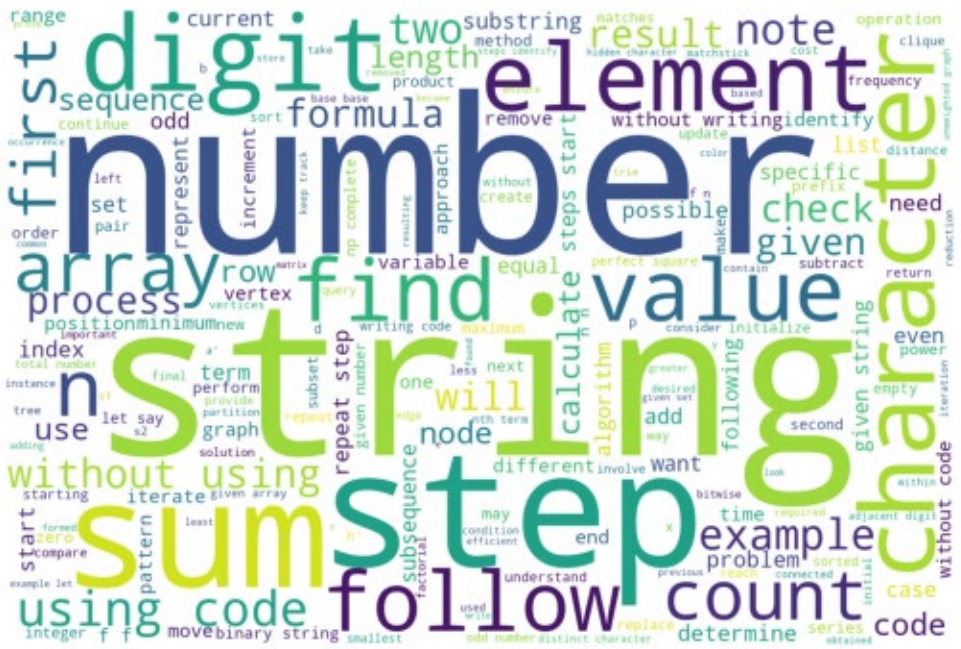

Figure 12: Word Cloud for Machine Generated text using Chat GPT 3.5

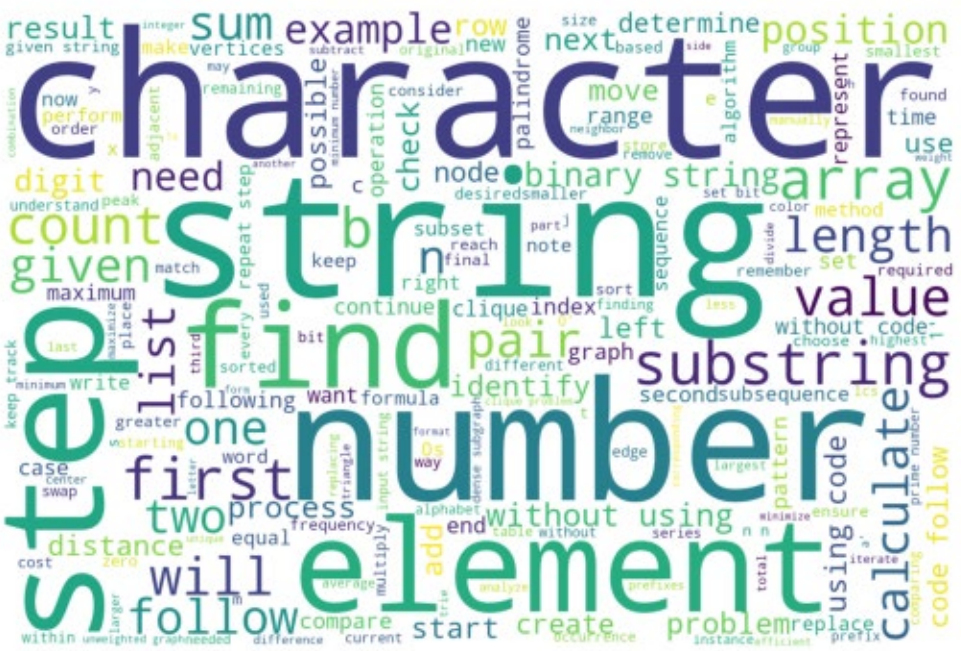

Figure 13: Word Cloud for Machine Generated text using Chat GPT 4.0

As we can visibly see from the word clouds that the words are similar for all the corpora. This shows that our method focuses on style and not on the words used by the authors.

4.6 Chapter Summary

4.6.1 Multimodal Stylometry

In this chapter, we were able to show the efficacy of multimodal stylometry (text and source code). We used a corpus with 19 authors and 50 documents per author for both text and source code. We employed the use of a robust feature selection process and use dearly stage fusion to combine the modalities. Table 22 showed that multimodality outperforms single mode stylometry across all the metrics that was used to evaluate the performance of the classifiers. We also explore the effects of document size on multimodality. Tables 23 to 34 presents results of 12 datasets with document sizes ranging from 10 documents per author to 70 authors per authors. Our scalability experiments show that Naïve Bayes and Random Forest perform very well for multimodality at the smallest number of documents (10) but Multilayer perceptron doesn't do well. However, we see a steady increase in the performances of MLP with multimodality as document size increases and, the results obtained from Random Forest and Naïve Bayes classifiers with multimodality experiences a steady improvement as document size increases. As document size gets to 55, we begin to see a decline in the performance of Random Forest with multimodality compared to single modality but Naïve Bayes and MLP results show an increase in multimodality performance as seen in Figures 8. 9 and 10. We also note that, though Naïve Bayes does not do

well with dimensionality, its performance doesn't seem to be affected when used for multimodality compared to when it's used for single modality. We also note that with most of the document sizes, the final fusion of features contained more features from the source code feature than they were from text features.

In conclusion, Random Forest classifiers yield the best results for small document size while MLP yielded the best results for larger document sizes with multimodality. However, Naïve Bayes maintained a better multimodality performance across all the document sizes that the single modalities.

4.6.2 Distinguishing human generated content from AI Generated Content

Finally, we explore the ability of our methodology to distinguish between human generated and AI generated content using Chat GPT 3.5 and Chat GPT 4.0 as our source for texts and source code generated from AI. Table 35 shows that our methodology performs well at making the distinction between human and AI generated texts and source code. Also, table 35 shows that multimodality outperforms single mode stylometry with Naïve Bayes classifiers and SVM but remains slightly unchanged with Random Forest and MLP classifiers.

In our pursuit of advancing stylometric analysis, we have undertaken a rigorous exploration into the distinctive characteristics that differentiate human-generated content from that generated by AI language models, specifically leveraging Chat GPT 3.5 and Chat GPT 4.0 as primary sources for data. Our focus extends beyond textual content to include source code generated by these AI models, providing a comprehensive examination of the capabilities of our methodology.

CHAPTER FIVE

CONCLUSION AND FURTHER WORK

5.1 Introduction

Stylometry is the study and analysis of the style (writing, painting, speech, writing code) of an individual. Multimodal stylometry is a stylometric technique that combines features from multiple modalities to improve accuracy. In this Chapter, we discuss the conclusions we draw from the experiments and results outlined in the previous chapter. The aim of these research was to show that multimodality in stylometry leads to better classification accuracy than single mode stylometry. We also show that machine generated text and source code (ChatGPT) can be distinguished from human generated text and source code.

5.2 Conclusion

The concept of modality is not new and has been applied in different fields with great success (Multimodal biometrics). In recent research, single mode stylometry has yielded good results for authorship identification and authorship profiling.

However, multimodality leverages on the best features from both modalities to improve accuracy. One major component of multimodality is the feature fusion. This is the technique of combining the information from both modalities.

In this work, our focus was to show that combining the features from the text and source code (multimodality) written by the same person improves classification accuracy when compared to single mode stylometry (text alone and source code alone). A major limitation we experienced was that there was no corpus available that had both text and source code written by the same author. To tackle this limitation, we identified some tutorial websites and used a python script to scrape the text and source code from the websites along with their authors. We also identified text stylometry feature set [6] and source code stylometry feature set [25] which were extracted from the text files and source code files. We employed the use of early fusion which is a technique that involves concatenating the features from the multiple modalities before building the model. Using machine learning classifiers, we built models using the individual feature sets and a third model using a combination of both feature sets (multimodality).

We evaluate our method by comparing the accuracy of all the models. The classification accuracy of multimodal stylometry showed to outperform the accuracy of the other models. We also carry out an experiment to distinguish between human generated text and source code and machine generated (ChatGPT) text and source code. Multimodality was also used. Our results showed great success in identifying text and source code generated using AI language models (Chat GPT 3.5 and Chat GPT 4.0). Although, unexplored in stylometry,

multimodality offers an improvement in authorship identification and can be very useful in the field of stylometry especially with ransomware attacker detection and academic plagiarism.

5.3 Future work

This work (Multimodal stylometry) is novel research that extends its influence across various domains, offering a multitude of avenues for expansion and application. Our work is not just a singular achievement but a gateway to a myriad of possibilities.

At the forefront of our endeavors lies the field of forensics, where the research conducted opens new dimensions in the identification and analysis of digital footprints. Unmasking the authors behind malicious software becomes a tangible reality as our research opens new methods for the detection of malware authors and ransomware attackers.

In the academic sphere, our work contributes to the field of plagiarism detection. By harnessing multimodality in stylometry, we empower educational institutions the ability to maintain the integrity of academic work especially in computer science/programming classes where students have to projects/assignments that comprises of both written text and source code. Multimodal stylometry could be useful, ensuring a level playing field and upholding the principles of academic honesty.

Our work not only addresses immediate applications but also serves as a catalyst for a deeper exploration into the unexplored field of multimodal stylometry. Beyond the specific applications outlined earlier, our endeavors open the door to an exciting array of avenues that beckon further investigation. One compelling avenue for extended research is in feature fusion techniques within multimodal stylometry. While our current work focuses on early-stage fusion techniques, the landscape of possibilities expands exponentially with the exploration of late-stage and hybrid feature fusion methodologies. This progression promises to enhance the sophistication of our applications, providing a more nuanced understanding of the intricate interplay between multiple modalities.

Also, the source code corpus is a canvas awaiting further exploration. In the pursuit of comprehensive understanding, we invite researchers to delve into the incorporation of multiple programming languages within the source code corpus. This expansion not only broadens the scope of our work but also lays the groundwork for a more inclusive and adaptable framework.

Finally, due to the unavailability of a multimodal dataset for stylometry which led us to undertake an extensive effort of text and source code acquisition from diverse online sources. This undertaking was instrumental in laying the foundation for our research. A larger and more robust corpus, comprising documents generated by real human authors across various domains would be very useful for multimodal stylometry. This corpus should be a comprehensive multimodal dataset that mirrors the complexity of real-world communication. This effort will not only fortify

the foundation of multimodality but will also catalyze advancements in the broader field of stylometry.

REFERENCES

[1] K. Alrifai, G. Rebdawi, and N. Ghneim, “Arabic tweeps gender and dialect prediction: Notebook for PAN at CLEF 2017,” *CEUR Workshop Proc.*, vol. 1866, 2017.

[2] M. Koppel, J. Schler, and K. Zigdon, “Determining an author’s native language by mining a text for errors,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 624–628, 2005, doi: 10.1145/1081870.1081947.

[3] M. Koppel, S. Argamon, and A. R. Shimoni, “categorizingTextByGender,” vol. 17, no. 4, pp. 401–412, 2002.

[4] H. Van Halteren, H. Baayen, F. Tweedie, M. Haverkort, and A. Neijt, “New machine learning methods demonstrate the existence of a human stylome,” *J. Quant. Linguist.*, vol. 12, no. 1, pp. 65–77, 2005, doi: 10.1080/09296170500055350.

[5] A. Nini, *A Theory of Linguistic Individuality for Authorship Analysis*. in Elements in Forensic Linguistics. Cambridge University Press, 2023. doi: 10.1017/9781108974851.

[6] E. Stamatatos, “A survey of modern authorship attribution methods,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009, doi: 10.1002/asi.21001.

[7] C. Suero Montero, M. Munezero, and T. Kakkonen, “Investigating the role

of emotion-based features in author gender classification of text," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8404 LNCS, no. PART 2, pp. 98–114, 2014, doi: 10.1007/978-3-642-54903-8_9.

[8] E. Dauber *et al.*, "Git Blame Who?: Stylistic Authorship Attribution of Small, Incomplete Source Code Fragments," *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 3, pp. 389–408, 2019, doi: 10.2478/popets-2019-0053.

[9] J. Kothari, M. Shevertalov, E. Stehle, and S. Mancoridis, "A Probabilistic Approach to Source Code Authorship Identification," 2007, pp. 243–248. doi: 10.1109/ITNG.2007.17.

[10] R. Hoshiladevi, P. Shireen, and P. Sameerchand, "Authorship Attribution Using Stylometry and Machine Learning Techniques," *Adv. Intell. Syst. Comput.*, vol. 384, no. January 2016, pp. 247–257, 2016, doi: 10.1007/978-3-319-23036-8.

[11] K. Nishiyama, G. O. Adebayo, and R. Yampolskiy, "Authorship Identification of Translational Algorithms," in *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 2021, pp. 90–91. doi: 10.1109/ICSC50631.2021.00023.

[12] D. Jagadiswary and D. Saraswady, "Biometric Authentication Using Fused Multimodal Biometric," *Procedia Comput. Sci.*, vol. 85, pp. 109–116, 2016, doi: 10.1016/j.procs.2016.05.187.

[13] P. Ravi, K. R. C. Babu, and J. A. Kumar, *Multimodal Biometrics for user*

authentication. 2017. doi: 10.1109/ISCO.2017.7856044.

[14] T. Agrawal, D. Agarwal, M. Balazia, N. Sinha, and F. Bremond, “Multimodal Personality Recognition using Cross-attention Transformer and Behaviour Encoding,” pp. 501–508, 2022, doi: 10.5220/0010841400003124.

[15] “PAN at CLEF.”

[16] J. A. Khan, “Author profile prediction using trend and word frequency based analysis in text: Notebook for PAN at CLEF 2017,” *CEUR Workshop Proc.*, vol. 1866, 2017.

[17] G. Kheng, L. Laporte, and M. Granitzer, “INSA Lyon and UNI passau’s participation at PAN@CLEF’17: Author Profiling task: Notebook for PAN at CLEF 2017,” *CEUR Workshop Proc.*, vol. 1866, 2017.

[18] D. Kodiyan, F. Hardegger, S. Neuhaus, and M. Cieliebak, “Author Profiling with bidirectional rnns using Attention with grus: Notebook for PAN at CLEF 2017,” *CEUR Workshop Proc.*, vol. 1866, 2017.

[19] L. Denoyer and P. Gallinari, “The Wikipedia XML Corpus,” *SIGIR Forum*, vol. 40, no. 1, pp. 64–69, Jun. 2006, doi: 10.1145/1147197.1147210.

[20] G. O. Adebayo and R. V Yampolskiy, “Automatic {IQ} Estimation from Written text using Stylometry Methods,” in *The 7th International Conference on Information System and Data Mining, {ICISDM} 2023, Atlanta, GA, USA, May 10-12, 2023*, ACM, 2023, pp. 56–65. doi: 10.1145/3603765.3603769.

[21] E. Stamatatos, G. Kokkinakis, and N. Fakotakis, “Automatic text

categorization in terms of genre and author," *Comput. Linguist.*, vol. 26, no. 4, 2000, doi: 10.1162/089120100750105920.

[22] Ramyaa, C. He, and K. Rasheed, "Using machine learning techniques for stylometry," *Proc. Int. Conf. Artif. Intell. IC-AI'04*, vol. 2, pp. 897–903, 2004.

[23] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler, "Automatically profiling the author of an anonymous text," *Commun. ACM*, vol. 52, no. 2, pp. 119–123, 2009, doi: 10.1145/1461928.1461959.

[24] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group Web forum messages," *IEEE Intell. Syst.*, vol. 20, no. 5, pp. 67–75, 2005, doi: 10.1109/MIS.2005.81.

[25] A. Caliskan-islam *et al.*, "De-anonymizing Programmers via Code Stylometry This paper is included in the Proceedings of the," *Proc. 24th USENIX Secur. Symp.*, pp. 12–14, 2015.

[26] H. Ding and M. H. Samadzadeh, "Extraction of Java program fingerprints for software authorship identification," *J. Syst. Softw.*, vol. 72, pp. 49–57, 2004.

[27] G. Frantzeskou, S. MacDonell, E. Stamatatos, and S. Gritzalis, "Examining the Significance of High-Level Programming Features in Source Code Author Classification," *J. Syst. Softw.*, vol. 81, no. 3, pp. 447–460, Mar. 2008, doi: 10.1016/j.jss.2007.03.004.

[28] G. Frantzeskou, E. Stamatatos, S. Gritzalis, and S. Katsikas, "Effective Identification of Source Code Authors Using Byte-Level Information," in

Proceedings of the 28th International Conference on Software Engineering, in ICSE ’06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 893–896. doi: 10.1145/1134285.1134445.

[29] E. Frank, M. A. Hall, G. Holmes, R. Kirkby, B. Pfahringer, and I. H. Witten, “Weka: A machine learning workbench for data mining.,” in *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, O. Maimon and L. Rokach, Eds., Berlin: Springer, 2005, pp. 1305–1314. [Online]. Available: http://researchcommons.waikato.ac.nz/handle/10289/1497

[30] M. Haghighat, M. Abdel-Mottaleb, and W. Alhalabi, “Discriminant Correlation Analysis: Real-Time Feature Level Fusion for Multimodal Biometric Recognition,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, pp. 1984–1996, 2016, doi: 10.1109/TIFS.2016.2569061.

[31] A. Selwal, S. Gupta, and S. Kumar, “A SCHEME FOR TEMPLATE SECURITY AT FEATURE FUSION LEVEL IN MULTIMODAL BIOMETRIC SYSTEM,” *Adv. Sci. Technol. Res. J.*, vol. 10, pp. 23–30, 2016, doi: 10.12913/22998624/64062.

[32] N. Hezil and B. Abdelhani, “Multimodal Biometric Recognition using Human Ear and Palmprint,” *IET Biometrics*, vol. 6, 2017, doi: 10.1049/iet-bmt.2016.0072.

[33] M. Hanmandlu, J. Grover, V. K. Madasu, and S. Vasirkala, “Score level fusion of hand based biometrics using t-norms,” *2010 IEEE Int. Conf. Technol. Homel. Secur.*, pp. 70–76, 2010.

[34] A. Ross, K. Nandakumar, and A. K. Jain, “Introduction to Multibiometrics,” in *Handbook of Biometrics*, A. K. Jain, P. Flynn, and A. A. Ross, Eds., Boston, MA: Springer US, 2008, pp. 271–292. doi: 10.1007/978-0-387-71041-9_14.

[35] D. Nauck, “Fuzzy Rule Learning With Symbolic Variables,” pp. 1–9.

[36] Y. Wang, D. Shi, and W. Zhou, “Convolutional Neural Network Approach Based on Multimodal Biometric System with Fusion of Face and Finger Vein Features,” *Sensors*, vol. 22, no. 16, pp. 1–15, 2022, doi: 10.3390/s22166039.

[37] V. Arulalan, V. Premanand, and G. Balamurugan, “An overview on multimodal biometrics,” *Int. J. Appl. Eng. Res.*, vol. 10, no. 17, pp. 37534–37538, Sep. 2015, doi: 10.5121/sipij.2013.4105.

[38] V. Rajasekar *et al.*, “Enhanced multimodal biometric recognition approach for smart cities based on an optimized fuzzy genetic algorithm,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, 2022, doi: 10.1038/s41598-021-04652-3.

[39] M. Castells, “The impact of the internet on society: A Global Perspective,” *Open Mind*, p. 25, 2014.

[40] M. Khonji, Y. Iraqi, and A. Jones, “An evaluation of authorship attribution using random forests,” in *2015 International Conference on Information and Communication Technology Research (ICTRC)*, 2015, pp. 68–71. doi: 10.1109/ICTRC.2015.7156423.

[41] M. Badar, M. Haris, and A. Fatima, “Application of deep learning for retinal

image analysis: A review," *Comput. Sci. Rev.*, vol. 35, 2020, doi: 10.1016/j.cosrev.2019.100203.

[42] A. P. Garibay, A. T. Camacho-González, R. A. Fierro-Villaneda, I. Hernandez-Farias, D. Buscaldi, and I. V. M. Ruiz, "A Random Forest Approach for Authorship Profiling," in *Conference and Labs of the Evaluation Forum*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:18882361

[43] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship Attribution with Support Vector Machines," *Appl. Intell.*, vol. 19, no. 1, pp. 109–123, 2003, doi: 10.1023/A:1023824908771.

[44] G. Anthony, H. Greg, and M. Tshilidzi, "Classification of Images Using Support Vector Machines," 2007, [Online]. Available: http://arxiv.org/abs/0709.3967

[45] P. Chhabra, R. Wadhvani, and S. Shukla, "Spam Filtering using Support Vector Machine," *Int. J. Comput. Commun. Technol.*, vol. 1, no. 4, pp. 256–261, 2010, doi: 10.47893/ijcct.2010.1053.

[46] X. Zhou, J. Li, C. Yang, and J. Hao, "Study on Handwritten Digit Recognition using Support vector machine," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 452, no. 4, pp. 0–6, 2018, doi: 10.1088/1757-899X/452/4/042194.

[47] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, "Gene Expression Data Classification Using Support Vector Machine and Mutual Information-based

Gene Selection," *Procedia Comput. Sci.*, vol. 47, pp. 13–21, 2015, doi: https://doi.org/10.1016/j.procs.2015.03.178.

[48] P. M. Shah, "Face detection from images using support vector machine," p. 23, 2012.

[49] S. Tian, J. Yu, and C. Yin, "Anomaly Detection Using Support Vector Machines," in *Advances in Neural Networks -- ISNN 2004*, F.-L. Yin, J. Wang, and C. Guo, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 592–597.

[50] W. Zhang and F. Gao, "An improvement to naive bayes for text classification," *Procedia Eng.*, vol. 15, pp. 2160–2164, 2011, doi: 10.1016/j.proeng.2011.08.404.

[51] H. Zhang and D. Li, "Naïve Bayes Text Classifier," in *2007 IEEE International Conference on Granular Computing (GRC 2007)*, 2007, p. 708. doi: 10.1109/GrC.2007.40.

[52] N. F. Rusland, N. Wahid, S. Kasim, and H. Hafit, "Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 226, no. 1, p. 12091, Aug. 2017, doi: 10.1088/1757-899X/226/1/012091.

[53] Y. Wang, J. Hodges, and B. Tang, "Classification of Web documents using a naive Bayes method," in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, 2003, pp. 560–564. doi: 10.1109/TAI.2003.1250241.

[54] S. Khan and M. R. Ghalib, “A naive-bayes approach for disease diagnosis with analysis of disease type and symptoms,” *Int. J. Appl. Eng. Res.*, vol. 10, pp. 29005–29014, 2015.

[55] C. Arouri, E. Mephu, N. S. Aridhi, C. Roucelle, G. Bonnet-Loosli, and N. Tsopzé, “Towards a constructive multilayer perceptron for regression task using non-parametric clustering. A case study of Photo-Z redshift reconstruction,” *Researchgate*, no. December, 2014.

[56] Y. Qin, C. Li, X. Shi, and W. Wang, “MLP-Based Regression Prediction Model For Compound Bioactivity.,” *Front. Bioeng. Biotechnol.*, vol. 10, p. 946329, 2022, doi: 10.3389/fbioe.2022.946329.

[57] S. S. Chai, W. L. Cheah, K. L. Goh, Y. H. R. Chang, K. Y. Sim, and K. O. Chin, “A Multilayer Perceptron Neural Network Model to Classify Hypertension in Adolescents Using Anthropometric Measurements: A Cross-Sectional Study in Sarawak, Malaysia.,” *Comput. Math. Methods Med.*, vol. 2021, p. 2794888, 2021, doi: 10.1155/2021/2794888.

[58] T. Bikku, “Multi-layered deep learning perceptron approach for health risk prediction,” *J. Big Data*, vol. 7, no. 1, p. 50, 2020, doi: 10.1186/s40537-020-00316-7.

[59] S. B. Cho, “Neural-network classifiers for recognizing totally unconstrained handwritten numerals.,” *IEEE Trans. neural networks*, vol. 8, no. 1, pp. 43–53, 1997, doi: 10.1109/72.554190.

[60] J. Serey *et al.*, “Pattern Recognition and Deep Learning Technologies,

Enablers of Industry 4.0, and Their Role in Engineering Research," *Symmetry (Basel).*, vol. 15, no. 2, 2023, doi: 10.3390/sym15020535.

[61] J. E. Casal and M. Kessler, "Research Methods in Applied Linguistics Can linguists distinguish between ChatGPT / AI and human writing ?: A study of research ethics and academic publishing," *Res. Methods Appl. Linguist.*, vol. 2, no. 3, p. 100068, 2023, doi: 10.1016/j.rmal.2023.100068.

[62] N. Islam, D. Sutradhar, H. Noor, J. T. Raya, M. T. Maisha, and D. M. Farid, "Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning," 2023, [Online]. Available: http://arxiv.org/abs/2306.01761

Appendix

1. The link contains files with the authors used and the topics from each of the authors.

   https://drive.google.com/file/d/11TRYi1Myx7kbHl59C03URROqQNsxU-m8/view?usp=sharing

CURRICULUM VITAE

NAME Glory O. Adebayo

ADDRESS 410 E Kenwood Dr,

Louisville, Kentucky. 40214

DOB Akure, Ondo, Nigeria – June 20, 1991

EDUCATION & TRAINING

Ph.D., Computer Science and Engineering
University of Louisville, Kentucky
2017-24

M.Sc., Computing: Information Engineering
Robert Gordon University, Scotland
2014-15

B.Sc., Computer Science
Covenant University, Nigeria
2007-12

PUBLICATIONS

- K. Nishiyama, G.O. Adebayo and R.V. Yampolskiy (2021), *Authorship Identification of Translational Algorithms* 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2021, pp. 90-91
- G.O. Adebayo and R.V. Yampolskiy (2022), *Estimating Intelligence Quotient Using Stylometry and Machine Learning Techniques: A Review,* in Big Data Mining and Analytics, vol. 5, no. 3, pp. 163-191
- G.O. Adebayo and R.V. Yampolskiy. (2023). *Automatic IQ Estimation from Written text using Stylometry Methods*. In Proceedings of the 2023 7th International Conference on Information System and Data Mining (ICISDM '23). Association for Computing Machinery, New York, NY, USA, 56–65.