Authorship Verification based on Linguistic Features

C. M. M. Dissanayake

Authorship Verification based on Linguistic Features

C. M. M. Dissanayake
Index No : 13000322

Supervisor: Dr. A. R. Weerasinghe

December 2017

Submitted in partial fulfillment of the requirements of the
B.Sc in Computer Science Final Year Project (SCS4124)

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: C. M. M. Dissanayake

……

Signature of Candidate Date:

This is to certify that this dissertation is based on the work of Ms. C. M. M. Dissanayake under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name: Dr. A. R. Weerasinghe

……

Signature of Supervisor Date:

Abstract

This thesis attempts to solve the problem of authorship verification. Authorship verification is a subdomain of authorship analysis and its origins lie in stylometry analysis. However most of the research in authorship analysis is based on authorship identification where authorship verification is rather unexplored. With the increase of digital documents and authors it is very difficult to employ authorship identification solutions. Hence in such cases authorship verification solutions are in necessity.

This research focuses on utilizing digital documents with 1000 words, written in English to solve the problem of authorship verification: coming into conclusion about the authorship of a text in dispute by analyzing texts written by some candidate author.

To solve this problem three machine learning models were designed employing two feature sets, comprising of linguistic features which are suggested to characterize the writing style of a person, one comprising of stylometric features and other consisting of word frequency based features. One-class support vector machine and two-class support vector machine are used as machine learning models to tackle this problem.

Results suggest one-class support vector machine with selected stylometric features does not tackle the problem very well while two-class classification model with stylometric features trained for known author class and unknown author class shows potential in solving the problem if the unknown author class can be properly represented. One-class support vector machine with word frequency based features, shows promising results in solving the authorship verification problem.

Preface

This research attempts to solve the authorship verification problem based on a machine learning technique. In that case a suitable feature set needs to be selected to model the writing style of a person. Hanlein's empirical research, has suggested such features and they are used frequently in literature to solve authorship identification problem. The proposed methods in this research tries to use the same feature set to solve the authorship verification problem using support vector machine as the classification model. Apart from that another set of features are selected based on frequent word usage and vocabulary levels of different authors and authorship verification problem is addressed. Even though support vector machines are often used in literature to solve the authorship verification problem, this would be the first time to employ the above mentioned feature sets to solve authorship verification problem using above mentioned techniques.

The experiments are designed and conducted by the researcher incorporating suggestions from the supervisor.

Acknowledgement

By conducting this research, I have developed an immense interest in stylometry analysis and natural language processing and gained my first experiences in research world. Hence I would like to thank my supervisor, Dr. A. R. Weerasinghe in introducing me to the project and advising me in any way needed and trusting my capabilities.

I would also like to thank Dr. H. Ekanayake for effectively coordinating the final year project in computer science and helping students in case of problematic situations. I am immensely grateful for my batchmates who also helped me in many ways and giving suggestions to improve the project. I would also like to be thankful towards my family who gave me great support and courage to carry out the research studies and providing with a suitable academic environment. Lastly my appreciation goes to everyone who helped during this attempt.

Table of Contents

List of Figures

List of Tables

List of Acronyms

SVM - Support Vector Machine

SVC - Support Vector Classification

AWL - Academic Word List

Definitions

Document - A digital file containing text entirely written by one person.

Known document - A document, with prior knowledge of the person who has written it.

Unknown document - A document, with no knowledge of the person who has written it.

Known author - The author who is suspected to have written the unknown document.

Unknown author - The author of the unknown document, if the unknown document is written by a different person than the known author.

Target class - The class that contain all documents with known authorship from a suspected author.

Outlier class - The class which contains all documents from other authors than the suspected author.

Chapter 1 - Introduction

Authorship analysis regards to analyzing characteristics of a document and coming into conclusions about its authorship and it is based on stylometry [1]. Research on analysis of authorship can be divided into three sub domains as authorship attribution, authorship verification and author profiling [1, 2]. Authorship attribution refers to identifying author from a pool of suspected authors by analyzing the writing style of the document. Authorship verification deals will assigning a label from "same author" and "not same author" to a document with unknown authorship with respect to documents obtained from a suspected author. Author profiling as suggested by the name attempts to predict a profile for the author including details such as age, gender, mother-tongue etc.

Stylometry is the application of authorship attribution by modeling the writing style learned from text of the documents [3]. Most common case of application of stylometry is for closed-world supervised learning: authorship attribution based models. But in most realistic problems these ideal scenarios are not present. Hence it requires semi-supervised open-world techniques: authorship verification methods.

This thesis is focused on exploring into authorship verification and authorship attribution backgrounds and to come up with a novel authorship verification model, based on machine learning techniques and stylometry.

1.1 Background to the Research

Complications regarding authorship can be dated back to medieval times [2]. With the importance of the document written, multiple people can claim to have written it. One such famous case is disputed authorship of twelve Federalist papers, studied by Mosteller and Wallace [4]. Revealing the authorship of a centuries-old Shakespeare play, "The Reign of Edward III" [5] shows how important it is to know the authorship of a certain document.

Stylometry which is an important interdisciplinary research area has focused on literary stylistics, statistics and computer science in studying the "style" or the "feel" of a document [6]. It assumes that there is a unconscious writing style to a writer that cannot be consciously manipulated over a short period of time, which retains distinctive features or qualities that enable in identifying a particular writer through his or her writing [6, 7]. Based on this rationale authorship analysis problems are formulated in attempting to model the writing style of an author.

Since authorship verification deals with verifying authorship from a given suspected author, the problem deals with open-world settings where there is prior information only on one candidate author through documents with known authorship and no information on any other possible authors. Due to that the problem is often pursued as a one-class classification problem, which is harder than a usual binary or multi-class classification problem [3].

1.2 Research Problem and Research Questions

Research Problem Statement

Given a text with authorship unknown and a few texts, of an author who is suspected to be the author of the authorship unknown text, verify whether the suspected author is same as the author of authorship unknown text.

Problem Description

The task of authorship verification is to come into conclusion about the authorship of the text in dispute by analyzing texts written by some candidate author. Given a suspicious document d and documents written by a candidate author $A = \{d_1, d_2, d_3 \ldots\ldots d_n\}$ the problem is to verify whether d also belongs to the set $\{d_1, d_2, d_3 \ldots\ldots d_n\}$ or not, as shown in Figure 1.1.

The underlying rationale behind authorship verification:

The writing style of an author is unique from person to person and it cannot be deliberately disguised over a short period of time and it can be used to verify the authorship of a document.

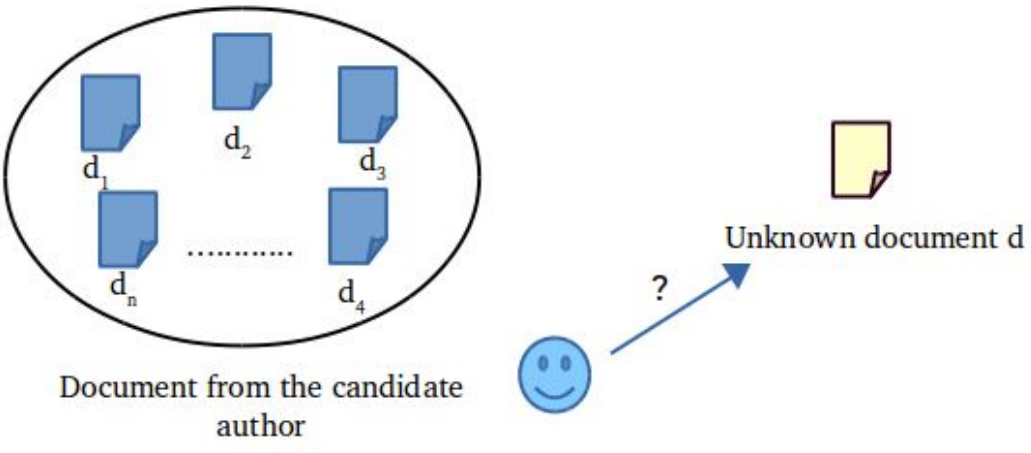

Figure 1.1 - Authorship verification problem [7]

Research Questions

Given the nature of the research problem following research questions are formulated to address different aspects of it.

1. Will the author be verified if document with unknown authorship is same as of documents with known authorship

2. Will the author be not verified if document with unknown authorship is not same as of documents with known authorship

3. How many authorship known documents are needed to create a successful model of the given author?

4. How robust the created model would be in case of imitation or obfuscation?

1.3 Justification for the research

With the increase of online communication and digital trends, digital documents such as emails, blogs, news articles etc. hold an important position in trustworthy communication. In such cases verifying the authorship of these documents is highly important in establishing this trustworthiness. When looking at forensic aspects authorship verification is of utmost importance where manual practices are currently employed. With the increase of use of digital documents for various purposes, and increase of cybercrime [1], it craves for automated forensic aspects of authorship verification practises. Current authorship verification methods can be utilized in pinpointing the right direction for an investigator in case of availability of less clues [8]. For trustworthy results and to find involved parties responsible or to present at a court of law, analysis methods needs to be improved further [9] and will not act as legitimate evidence.

Due to these urgencies research has been conducted in similar domains. Even-though research in computational stylometry is ample [3], the one-class classification of authorship verification is rather unexplored [10]. Hence the need to improvement in this research area is plentiful in accuracy wise and in other aspects such as purpose specific.

1.4 Methodology

In this research study, it is intended to come up with a novel authorship verification model by studying existing models and authorship analysis domain research. After an intensive literature review, gaps are identified and a model is hypothesized according to research needs.

The main methodology used in this study is quantitative research approach. The first step in creating a authorship verification model would be feature selection and extraction. According to authorship attribution, authorship verification and linguistic studies, features

which can uniquely identify an author are selected and they are extracted. Further details on this process are examined in sections 2.2 and 3.1.

After determining the feature vector, it is important to determine a suitable computing technique for classification. Machine learning models are examined through the literature and a suitable model is selected according to features extracted.

Next important step is to train, test and evaluate the model. Targeting the data available, a necessary sampling technique is employed to sample data and train and test the model. After training, the model will output the label “same” or “not same” when documents from suspected author and document with unknown authorship are given as inputs. Detailed processes are discussed in chapter 3. The evaluation is carried out from testing outputs regarding suitable evaluation measures.

1.5 Delimitations of Scope

This study only focuses on documents written in English language. Languages like Sinhala is not considered due to scarcity of finding documents written in Sinhala suitable for the nature of the research. Other languages are not considered because of researcher’s lack of familiarity with them.

Documents around thousand words are considered for authorship verification process and processing shorter texts is not considered. This is because sufficient amount of data needs to processed to model a writing style for a person which can be distinguished from others. When considering a document with unknown authorship for authorship verification task, there should be multiple documents from the suspected author to increase the success rate of the verification.

1.6 Outline of the Dissertation

Sections in chapter 1 will present definitions, limitations of scope and conclusion for chapter 1. Through chapter 2, a thorough literature review of the domain of authorship verification is presented, incorporating authorship attribution and stylometry areas. In chapter 3 research design is presented with detailed descriptions of feature extraction, machine learning models. In chapter 4 experiment implementations are given in detail which are formulated to address given research questions. In chapter 5 results and evaluation details are mentioned regarding the experiments conducted. At the end from chapter 6 conclusions and contributions from the research and future research work are described.

1.7 Summary

From the above sections the research background is related, how the authorship verification research is conducted and cases which are solved successfully thanks to research in this field. Then the research problem is briefly described. The formulated research questions are presented, which are based on the nature of the research problem. Definitions used in research are given, and the need for research is justified with examples and research methodology is presented concisely. The dissertation is outlined and limitations in scope are given with reasoning. On this basis, the dissertation can continue with a detailed overview of the research.

Chapter 2 - Literature Review

Authorship verification in the context of stylometry can be referred to as modelling writing style of a given author to identify authorship of an unknown authorship document, same or not [10]. Stylometry is the authorship analysis by learning the style of a document. It has existed for centuries with historical, literary and forensic applications [3]. The most famous application of stylometry study in the history is the authorship verification between candidate authors James Madison or Alexander Hamilton on 12 Federalist Papers [38]. Over the years many research has been conducted concerning authorship analysis tasks. Among them most of them are focused on authorship attribution task and authorship verification task is relatively unexplored due to its open-world nature [10].

Following sections in this chapter will further explore into underlying background theories in data sampling, feature extraction and computational models in the related areas of authorship analysis tasks. Figure 2.1 gives an overview of the background theories.

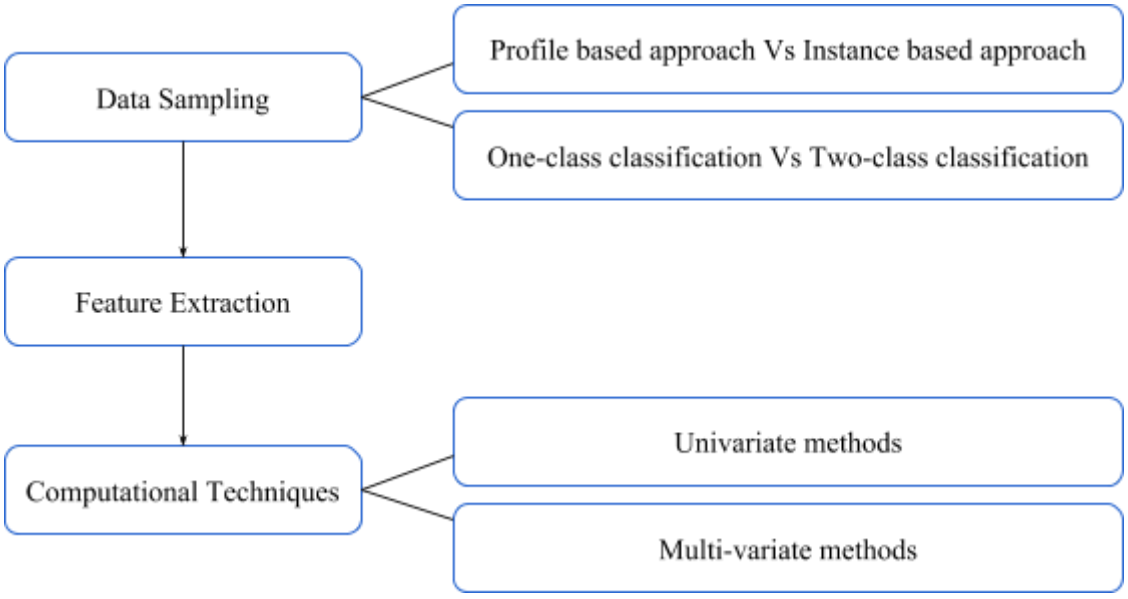

Figure 2.1 - Overview of the background theories [7]

2.1 Data Sampling

In this section different methods of authorship attribution, authorship verification and related methodologies are classified according to the data sampling method utilized by each.

2.1.1 Profile based approach Vs Instance based approach

There are many authorship attribution methods, and Stamatatos [13] classifies all these methods as profile based approaches and instance based approaches. In profile based approaches all the known documents of an author are concatenated together to create a profile. Features are extracted from the concatenated text and the profile is created and it is used to identify the most likely author for the unknown document based on a distance measure.

Profile based approaches consist of compression and probabilistic models [7, 13]. However these approaches are often criticized for losing information as all the dissimilar content is removed from text when profile is created [13]. Figure 2.2 shows the architecture of a typical profile based authorship attribution method.

Most of the modern authorship attribution methods utilize instance based approaches. In such cases each known document from a author are considered as an instance of the problem [7, 13]. Instance based methods can retain all the details from the documents and the classification model is trained from each instance of the text. In these cases data sample length should be adequate to represent writing style of an author [13].

Instance based approaches majorly consist of vector space models, but similarity-based and meta-learning models are also employed [13]. Below Figure 2.3 shows a common architecture for instance based approaches.

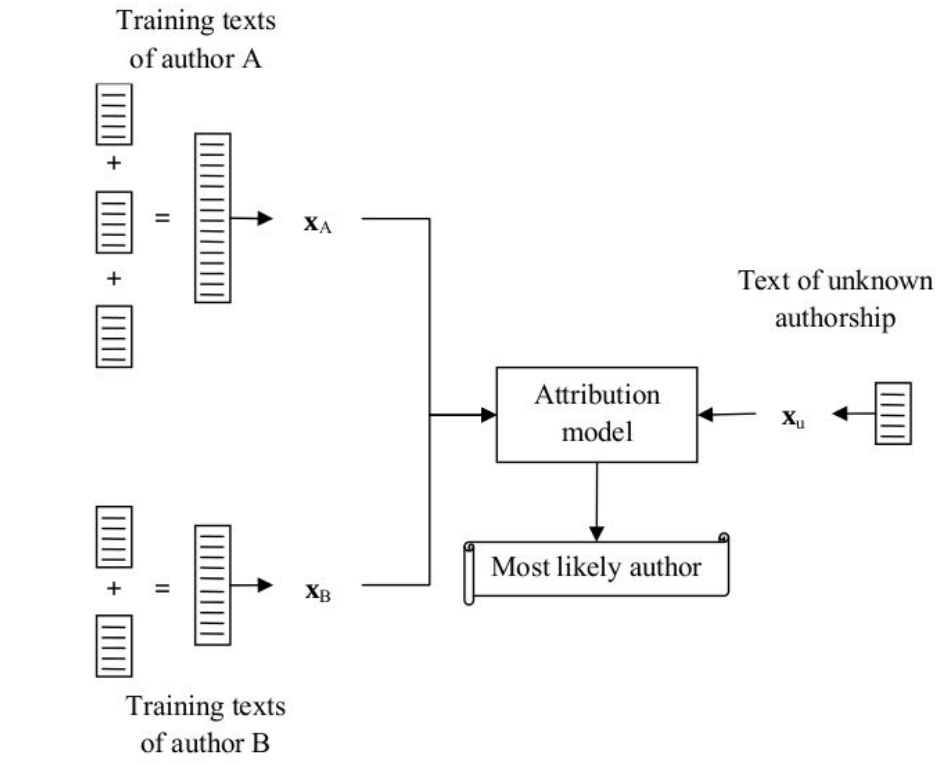

Figure 2.2 - Architecture of profile based approaches [13]

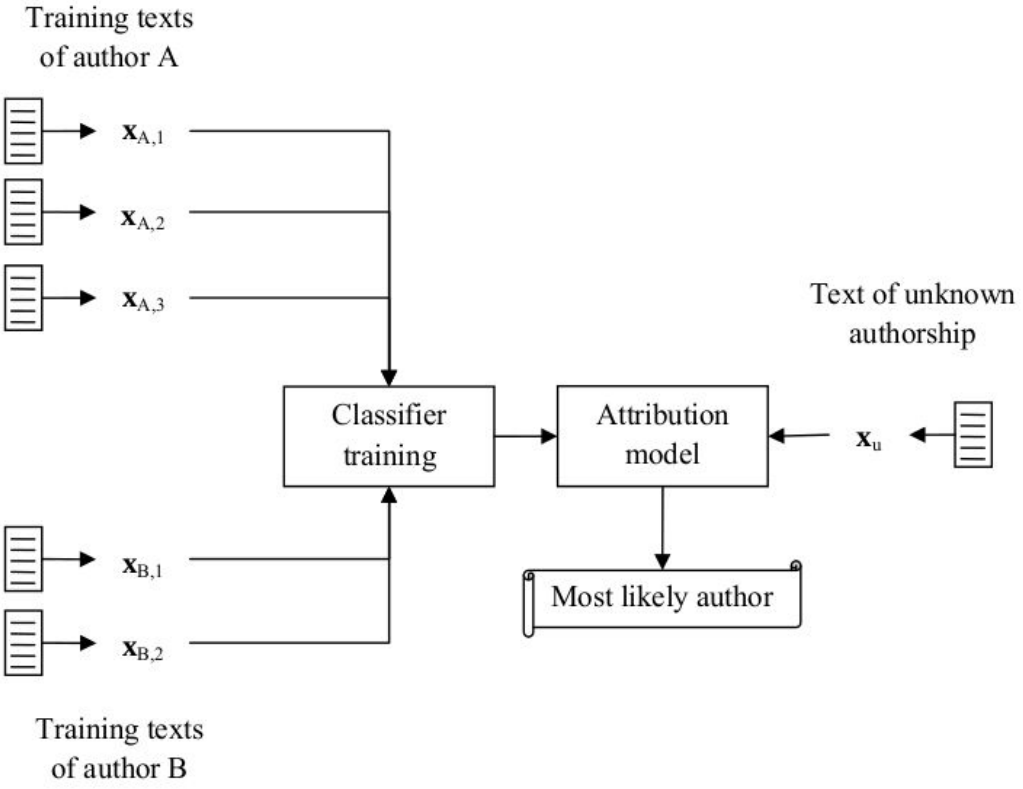

Figure 2.3 - Architecture of instance based approaches [13]

2.1.2 One-class classification Vs Two-class classification

One-class classification can be described as checking whether an object belongs to a considered class only [13]. If an object does not belong to the considered class, then it can belong to any other class hence we consider them as outliers. As shown in Figure 2.4 the target class represents the considered suspected author while all other authors are considered as outliers in the context of authorship verification problem.

One method in the one-class classification is to reduce the problem to binary by generating outliers around the target sample set. But this requires outliers closer to the target for a better classification. Three common ways to approach one-class classification are density estimators, reconstruction methods and boundary methods [16]. Density estimators consider that the outlier data is uniformly distributed and it directly estimates the probability distributions of the target class features while reconstruction methods make use of prior knowledge and assumptions about the generation process to fit a model to the given data. Boundary methods focus on creating a boundary among target set. One feature shared by all these methods is usage of a distance measure from the object to the target class which acclaims to the resemblance of the object with the target.

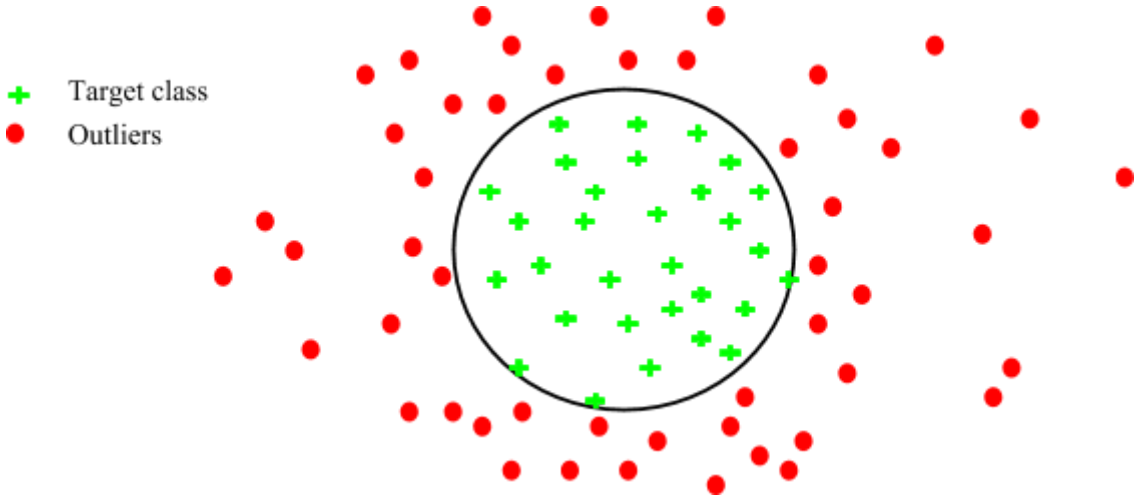

Figure 2.4 - One-class classification: class boundary representation

Additional properties of these methods include: robustness to outliers, ease of configuring parameters, computation and memory requirements.

When considering one-class classification methods Support vector machine (SVM) classifiers are proven to be effective in most cases [3]. However methods such as Bayesian classifiers, linear discriminant analysis, neural networks, and decision trees has become useful in some instances. Unmasking algorithm presented in [14] and baseline distractorless authorship verification framework given in [17] shows the most potential in solving the authorship verification problem when considered as a one-class classification problem.

However when the documents from a suspected author becomes limited the one-class classification approach becomes rather biased and results may get unreliable. Therefore two-class classification approach can be employed. In such cases outlier class can be created in such way that it contains discriminating features from the target class. Hence outlier class needs to be properly represented and it needs to be closer to target class for better classification [7] as shown in Figure 2.5. This is because if an outlier class far away from target class is selected, then an unknown document closer to target class but actually does not belong to target class is given, there is a high chance of misclassification. This scenario is depicted in Figure 2.6.

2.2 Features

Modern authorship attribution methods originated from Mosteller and Wallace's work in 1964 to solve dispute in twelve Federalist papers [2]. This was done by adopting distributions of function words as a discriminating feature to settle the disputed authorship between suspected authors [11]. The distributions of function words and syntactic features act as good markers of unconscious writing style and hence provide good clues on authorship [11]. Syntactic structures [12, 13], n-grams of syntactic labels from partial parsing [28], n-grams of parts-of-speech [14], complexity and richness measures (such as sentence length, word length, type/token ratio) and functional lexical features [15] have all been claimed to be reliable markers of style.

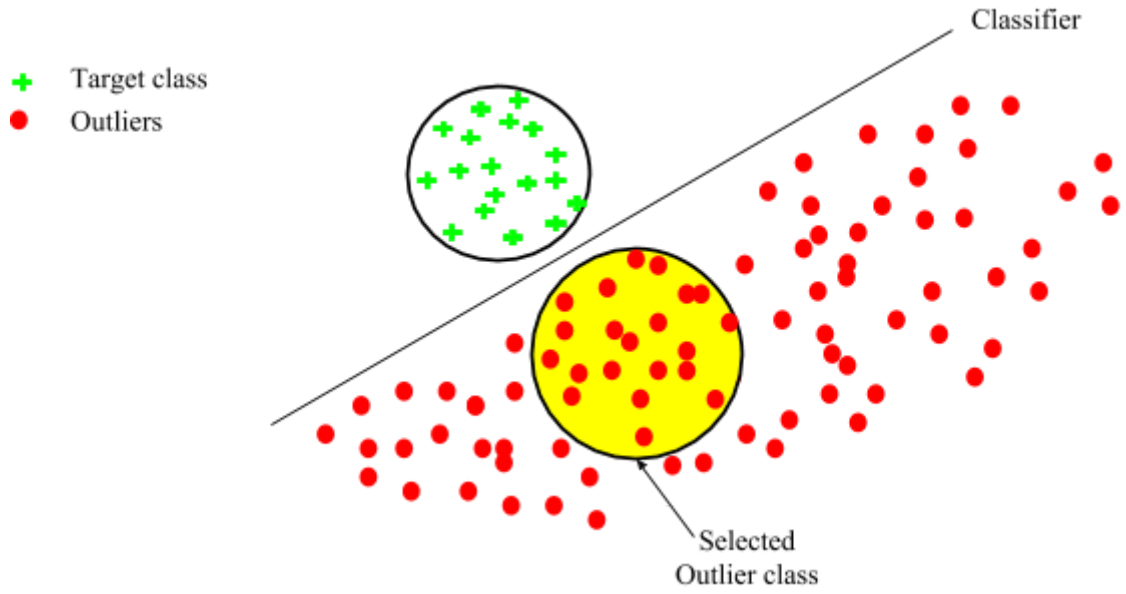

Figure 2.5 - Two-class classification: with proper outlier class selection

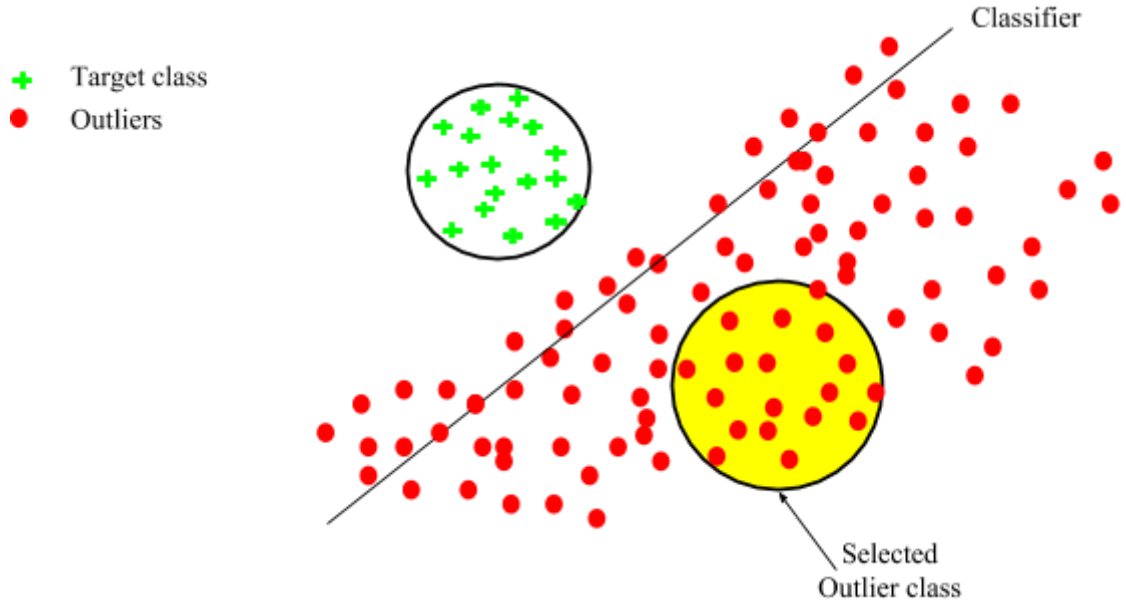

Figure 2.6 - Two-class classification: with improper outlier class selection

Corney et al. [20] show that the most successful features are the function words and character n-grams whereas McCombe et al. performed the tests using word uni-grams as classification feature, for which the results were promising [2, 21]. Hirst et al. [22] used tag bigrams to distinguish the writing of two authors with three experiments.

There is no standard discipline to determine whether which features needs to be used in stylometry analysis. Many different types and combinations of features have been used in various experiments before. Glover et al. [18] presents a comprehensive table of features shown in table 2.1 which are used in author identification studies which are supposed to model a person's writing style.

2.3 Computational Techniques

Authorship analysis tasks are often carried out employing techniques which are univariate or multivariate statistics. Multivariate statistics consists of machine learning techniques. Further details are discussed below.

2.3.1 Univariate Approaches

The origin of the scientific approaches for authorship attribution can be dated back to nineteenth century in the work of Mendenhall and Mascol [2]. The idea proposed in these approaches is that writing style of an author can be characterized by a unique curve expressing the relationship between word length and relative frequency of occurrence [2]. This led to searching for invariant properties in textual properties in the early twentieth century [2]. But however these methods are proven to be unstable, hence paving way for multivariate approaches [19].

Table 2.1 - Table of features used in authorship attribution studies [18]

Unanalyzed text	**Tagged text**
• Register of words used (formal, slang, technical, etc) • Frequent words (at least 3 per thousand) • Sentence length (mean and standard deviation) • Word length (mean and standard deviation)	• Type / token ratio • Distribution of word classes (parts of speech) • Distribution of verb forms (tense, aspect, etc) • Frequency of word parallelism • Distribution of word-class patterns (e.g., determiner + noun + verb) • Distribution of nominal forms (e.g., gerunds) • Richness of vocabulary
Parsed text	**Interpreted text**
• Frequency of clause types • Distribution of direction of branching • Frequency of syntactic parallelism • Distribution of genitive forms (of and ’s) • Distribution of phrase structures • Frequency of imperative, interrogative, and declarative sentences • Frequency of topicalization • Ratio of main to subordinate clauses • Distribution of case frames • Frequency of passive voice	• Frequency of negation • Frequency of deixis • Frequency of hedges and markers of uncertainty • Frequency of semantic parallelism • Degree of alternative word use (preference for synonyms) •

2.3.2 Multivariate approaches

Origin for multivariate approaches can be considered as Mosteller and Wallace's work for disputed twelve federalist papers, in which Bayesian classification is employed on frequencies of a set of a few dozen function words [2].

A basic intuition based in these methods is to view groups of texts from each author as points in some space and assigning the document in question to the group which is closest to it using some distance measure [2]. Burrow's Delta method can be considered as one such method which is used for many number of authorship attribution works [23, 24, 25].

With the emergence of text categorization techniques, based in machine learning, marked an important turning point in authorship attribution models. In such cases the application of such models is very straightforward where documents from authors are represented as numerical vectors and boundaries are demarcated between classes of authors [2]. Thus statistical and machine learning techniques such as Discriminant Analysis, Support Vector Machines, Decision Trees, Neural Networks, Genetic Algorithms, Memory-based learners, classifier ensemble methods can be employed to train classification models [13].

Koppel et al. mention after an extensive literature survey that support vector machines are at least as good for text categorization as any other learning method and the same has been found for authorship attribution [2]. The same study shows that some variations of Winnow and Bayesian regression also gives promising results in text categorization [2].

Furthermore Koppel et al. [14] consider the problem as a one class classification problem and has introduced an unmasking algorithm which uses a SVM classifier to do the classification which shows promising results. A supervised learning technique adopted by Brocardo et al. [26] combined with n-gram analysis and gains a EER of 14.35%. Most recently Mechti et al. [27] creates a hybrid authorship verification model by combining linguistic features and n-gram analysis and using SVM as the learning classifier.

Support Vector Machines

Support Vector Machines are one of most famous machine learning algorithms, often used in literature for supervised learning tasks. SVM is based on the Structural Risk Minimization Principle in Computational Learning Theory [48]. The Structural Risk Minimization Principle suggests to find a hypothesis h for the lowest true error. The true error of h is the probability that h will give an error for an unseen and randomly selected test sample. An upper bound is introduced to connect true error of hypothesis h to error of h on the training set and the complexity of H [48]. SVMs find the hypothesis h which minimizes this bound on true error [48].

SVM's behavior can be changed by using different kernel functions. The most popular kernel functions are :

1. The linear kernel
2. The polynomial kernel
3. The RBF (Gaussian) kernel
4. The string kernel

It is often recommended to use linear kernel for text categorization tasks since most of the text data are linearly separable [48]. Also utilizing a linear kernel is faster and only requires only a few parameters to be optimized.

2.4 Summary

Over the history there has been many cases of disputes regarding authorship. Since written documents play an important role in communication and expressing ideas, it is vital to know the originator of each, before serving their true purpose. It has been attempted to solve such authorship related problems over the course of research through manual analysis [4]. But with the advancement of computational techniques and algorithms, focus has been applied to automated methods in authorship analysis.

Since there has been many cases where manual authorship analysis consumes time and effort, this has become a necessity. One such famous case is the case of "UNABOMBER", where a man who was responsible for bombings were identified by a 35,000 word document sent by him to the FBI, analyzing it with the previously written letters and documents by the suspect [33]. At that time the resources and techniques utilized were limited on authorship analysis, but the evidence on arrest was mainly based on the linguistic analysis and the suspect was put away [33]. Authorship analysis in a forensic aspect has become immensely helpful in such way and automated authorship analysis techniques hold more potential importance in helping the society.

Authorship analysis techniques employed so far contain univariate and multivariate techniques [35][36] such as support vector machines, decision trees [13] and frequent pattern mining [37]. However, there is still need for improving feature sets used and techniques employed that can be trustworthy to a degree, where involved parties can be held responsible or presented in a court of law [9].

Chapter 3 - Design

The authorship verification model design can be done by considering the problem as a one class classification problem or two class classification problem. The difference in these two approaches is in selecting the outlier class where one class classification approach does not require selecting an outlier class while two class classification approach depends on how the outlier class is selected. The idea of one class classification is to use the most of the authorship known documents and to create a boundary for target class to determine the label of the authorship unknown document [7]. Three models are created, two one class classification models and a two class classification model, utilizing below described different feature sets.

3.1 Feature selection

3.1.1 Feature set I

After considering features often used in the literature, proven to characterize the writing style of a person and Hanlein's empirical research and based on work of Rasheed et al. [28, 29] below stylometric features are selected which indicate different styles of writing.

1. type-token ratio: The type-token ratio indicates the diversity of an author's vocabulary.
2. mean word length: Longer words indicate the formality of the writing style, while shorter words are a typical feature of informal spoken language.
3. mean sentence length: Sentence length indicates whether the writing is done carefully, planned or whether it is informal.
4. standard deviation of sentence length: The standard deviation shows the variation of sentence length.
5. mean paragraph length: The paragraph length is much influenced by the occurrence of dialogues.

6. chapter length: The length of the sample chapter.

7. number of commas per thousand tokens: Commas signal the flow of ideas within a sentence.

8. number of semicolons per thousand tokens: Semicolons shows the reluctance of an author to stop a sentence at some points which he/she could.

9. number of quotation marks per thousand tokens: Frequent use of quotations is considered a typical involvement feature [29].

10. number of exclamation marks per thousand tokens: Exclamations shows strong emotions.

11. number of hyphens per thousand tokens: Could help in uniquely identifying a author some others.

12. number of occurrence of word “and” per thousand tokens: And act as marker of coordination.

13. number of “but” per thousand tokens: The contrastive linking but also indicate coordination.

14. number of “however” per thousand tokens: The conjunction “however” is meant to form a contrastive pair with “but”.

15. number of ifs per thousand tokens: If clauses are samples of subordination.

16. number of word “that” per thousand tokens: Most of the time “that” is used for subordination while a few are used as demonstratives.

17. number of “more” per thousand tokens: “More” is an indicator of an author’s preference for comparative structure.

18. number of times “must” occur per thousand tokens: Modal verbs act as potential candidates for expressing tentativeness [29]. “Must” is more often used non epistemically.

19. number of “might” per thousand tokens: “Might” is more epistemically used.

20. number of “this” per thousand tokens: “This” is typically used for anaphoric reference.

21. number of “very” per thousand tokens: “Very” is stylistically significant for its emphasis on its modifiees.

3.1.2 Feature set II

Intrinsic plagiarism detection: detecting plagiarized passages automatically, within a document with no reference documents by analyzing deviations in writing style is a sub-task of authorship analysis domain [42]. According to Eissen et al. even human readers can identify potential plagiarisms by examining the document: changes between brilliant and baffling passages, or based on the change of person narrative [42]. Eissen et al. also point out the customariness of word usage can significantly capture a part of writing style [42]. Research has shown that the most frequent 9000 word families with proper nouns provide coverage of over 98% of the running words in a wide range of texts [43]. According to [44] learning of English words requires repetition and they gradually learn the most frequent 9,000 word families. Hence the vocabulary richness of a person can be analysed through examining the frequencies of words used, according to frequency rankings of the words in most frequent 9000 words.

Hence for the feature set II, word frequency based features are selected. Longman Communication 9000: is a list of the 9000 most frequent words in both spoken and written English, based on statistical analysis of the 390 million words contained in the Longman Corpus Network [45]. These 9000 words are divided into three subsets of size 3000 as high frequency words, mid frequency words, lower frequency words [46]:

1. high frequency words – indicates the top 3,000 words
2. mid frequency words – indicates the next most important 3,000 words
3. lower frequency words – indicates the less frequent yet important 3,000 words

Apart from that, word frequencies from the Academic Word List (AWL) [47] can also be considered when deviating among different vocabulary constituents. The AWL is a list of 570 word families that are commonly found in academic texts [47]. Therefore it will serve as an indicator in case the writer has any academic influences.

Hence the feature set II contains below features:

1. Percentage of high frequency words in the document
2. Percentage of mid frequency words in the document
3. Percentage of lower frequency words in the document
4. Percentage of words in document present in AWL.

3.2 Feature extraction

Above identified feature sets I and II are extracted separately from the sample training and testing data sets and converted to feature vectors.

Feature vector for a known document can be represented as V = [f1, f2, f3, ……., fn] where n is the number of features extracted. Figure 3.1 shows the feature matrix V_k, generated by combining feature vectors of all known documents.

$$\begin{pmatrix} f11, f12, f13, \ldots\ldots\ldots, f1n \\ f21, f22, f23, \ldots\ldots\ldots, f2n \\ f31, f32, f33, \ldots\ldots\ldots, f3n \\ \vdots \\ fm1, fm2, fm3, \ldots\ldots\ldots, fmn \end{pmatrix}$$

Figure 3.1 - Feature matrix extracted from known documents

Similarly, a feature vector from an unknown document is extracted such as V_u = [f'1, f'2, f'3, ……., f'n]. V_k and V_u can be used to test the classification model after trained by feature matrix created from known documents, hence training accuracies are obtained.

Same is repeated for testing data and testing accuracies are found. Further details on classification model is given in 3.2.

3.2 Classification Model Design

3.2.1 One-class classification model design

According to the research methodology described in Section 1.4, a classification model as shown in figure 3.2, which is based on machine learning is created. A one class support vector machine (SVM) classifier is trained with a linear kernel. SVM classifier is chosen as it is often used in the literature and has outperformed other classifiers such as decision trees, nearest-neighbors, Bayesian classifiers and so on [7, 14, 27]. The classifier after training, takes authorship known documents from testing data and predicts specific label (“Y” , ”N”) for the authorship unknown document.

3.2.2 Two-class classification model design

The classification model is trained for two classes such as known author and unknown author. Since documents are obtained from the known author, features extracted from those are used for training for the known author class. The only data available to indicate the unknown author is the unknown document. Hence it is very difficult to model a class for the unknown author. Since the variations of writing styles of different writers are very minute, the unknown class needs to be trained in such way that it is very closer to the known author class.

From the experiments conducted before, by thinking the model as an authorship attribution model and using the same features mentioned in section 2.2, it was evident that model is capable of distinguishing between two authors when the model is trained for two classes with features extracted from documents obtained from given authors. Hence the feature set employed showed the capability to distinguish between different writing styles. Unknown author is modelled by getting mean values of each feature extracted from known author documents and each value would deviate minutely. Such vectors with minute deviations are used in training of the unknown author class.

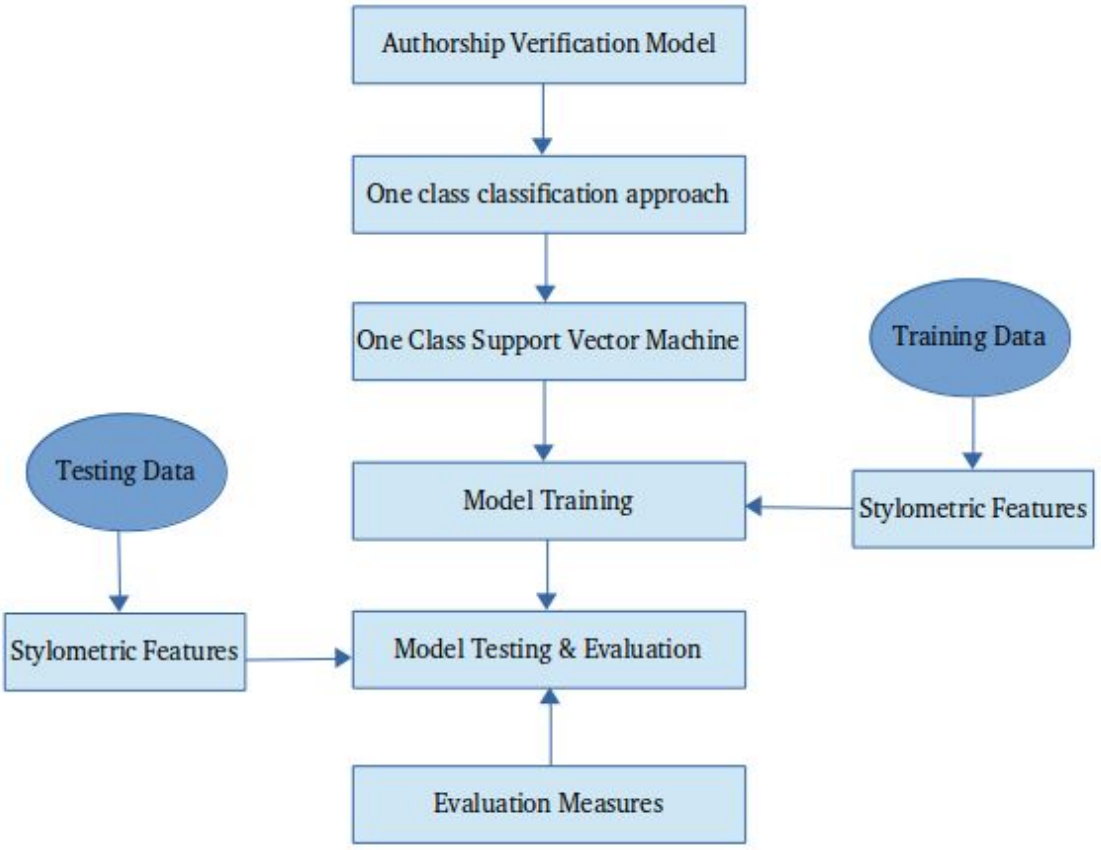

Figure 3.2 - Authorship verification research design

After training the model for two classes with features extracted from known documents and prepared vectors for unknown class, given unknown document is given as input to the system for testing. Features are extracted from the unknown document and model would output “known” or “unknown” label indicating the document is written by the same author as known documents or not. Training accuracies can be obtained in this training phase. Same process is carried out in testing phase and testing accuracies are obtained.

3.3 Data

3.3.1 Data collection

The data set which is used for the research would be from PAN @ CLEF competition where they provide data sets used for 2013, 2014 and 2015 years [30]. Apart from that

"Reddit Cross-Topic Authorship Verification Corpus" will be used which consists of documents generated from 1000 users [31]. Reuters Corpus Volume 1 [39, 40] is also used adapted for authorship verification requirements. When setting up experiments Extended-Brennan-Greenstadt Adversarial Corpus [32] is used to detect how classifier would behave when author tries to imitate the writing style or deliberately tries to change the writing style.

3.3.2 Data preprocessing

The documents provided by PAN corpus [30] are given as plain text and they are encoded into UTF-8 format. There are separate training and testing datasets given. Then they are converted to lowercase for easier parsing and text is split and converted to tokens as word tokens, sentence tokens and paragraph tokens for feature extraction. Another .txt file contains the truth about each case as "Y" or "N" whether it is the same author or not, of the document sets along with the names of the folders the document sets are residing in. These values are read from the file and stored in a separate vector for document reading and model evaluation. Complete flow of preprocessing steps are shown in Figure 3.3.

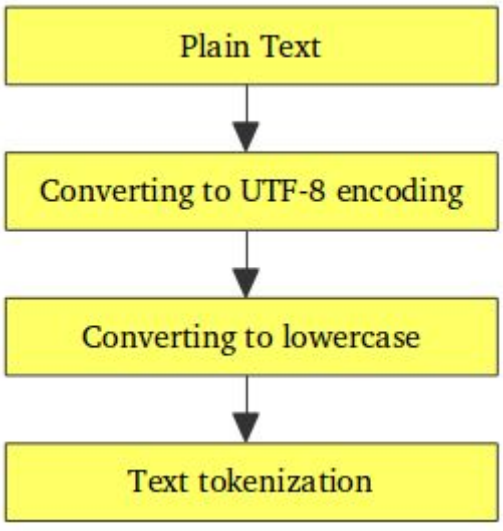

Figure 3.3 - Steps in data preprocessing

3.4 Experimental Design

Experiments were setup to address research questions in Section 1.2 by implementing described classification models with feature sets I and II. Hence three such models were created.

1. One-class classification model with Feature set I
2. One-class classification model with Feature set II
3. Two-class classification model with Feature set I

Two-class classification model with Feature set II was not created since the number of features in feature set II is not enough to create a proper representation for outlier class in two-class model.

3.4.1 Experiment design to address research question 1[1] and 2[2]

3.4.1.1 Model I: One-class classification model with Feature set I - experimental design

A document set would contain multiple documents with known authorship from some author and a document with authorship unknown. There are multiple sets with such cases. Document sets with unknown authorship same as known authorship and document sets with unknown authorship different than known authorship are selected randomly and created a data set containing 50 document sets as training set while 50 document sets as testing set. Hence 50 authors are subjected training and testing. From this set of 50 authors 3 datasets are created with 10 document sets such that:

[1] Will the author be verified if document with unknown authorship is same as of documents with known authorship ?

[2] Will the author be not verified if document with unknown authorship is not same as of documents with known authorship ?

1. In all cases the unknown author is same as suspected author
2. In all cases the unknown author is different than suspected author
3. Five random cases with unknown author being same as suspected author and five random cases with unknown author being a different author

Next a training and testing set of 200 document sets are created randomly from PAN dataset to evaluate the model.

These are fed to the verification model created, where it will extract aforementioned 21 stylometric features and would create a feature vector with dimensions, number of features x number of documents in one instance (n), (21 x n) with respective labels. 50 such feature vectors are created for each document set and a one class SVM is trained and tested for each instance. The results are interpreted with real labels of the document sets ("Y" , "N"). Then the experimental results are analyzed. Figure 3.4 shows the steps in the experimentation.

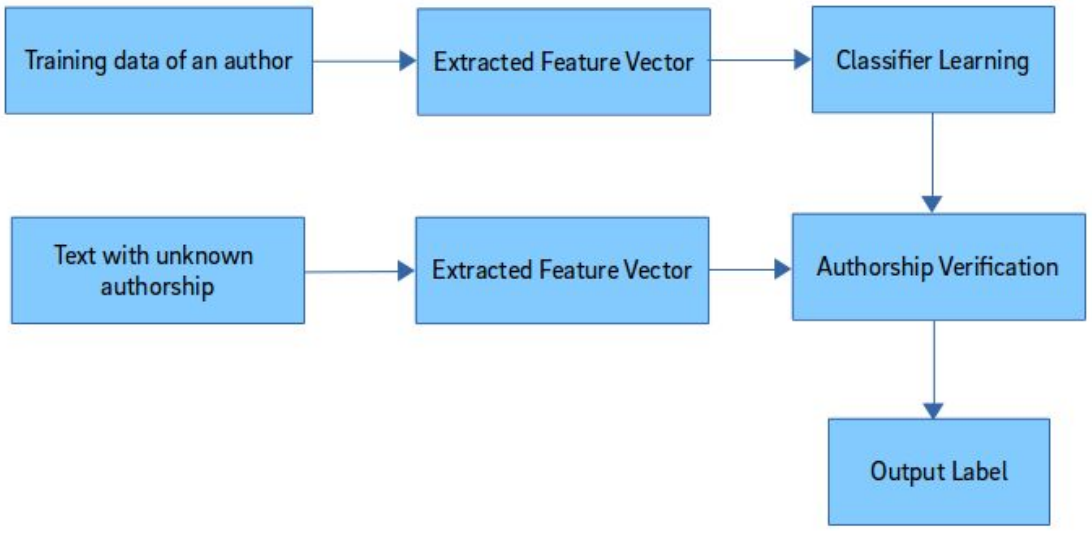

Figure 3.4 - Experiment design for research question 1 and 2 for one-class classification model

3.4.1.2 Model II: One-class classification model with Feature set II - experimental design

In the same way, as mentioned above in Section 3.4.1.1, a document set would contain multiple documents with known authorship from some author and a document with

authorship unknown. Document sets with unknown authorship, same as known authorship and document sets with unknown authorship, different than known authorship are selected randomly and created a data set containing 50 document sets as training set while 50 document sets as testing set. Then there would be 25 cases with authorship same as suspected author and 25 cases with authorship different from suspected author.

Feature sets are created the similar way as in 3.4.1.1 with dimensions, number of features x number of documents in one instance (n), (4 x n) from the above created training and testing sets.

3.4.1.3 Model III: Two-class classification model with Feature set I - experimental design

The dataset contains 50 authors with training data and testing data randomly selected. Each author contains 50 documents for each training and testing datasets. Dataset is modified such that 25 authors would contain an unknown document written by the same author and other 25 would have an unknown document written by a different author. 21 mentioned features are extracted from each known document for an author and labels are assigned for each feature set as "known". Then the feature vectors which are designed to represent the unknown class are combined with known class feature vectors and feature matrix is created. Then the SVM is trained with feature matrix for two classes "known" and "unknown". Afterwards features are extracted from the given unknown document and label is predicted for the document using the trained model. Same process is repeated for each author and training accuracies are calculated. Then the process is repeated for testing data and model is evaluated.

3.4.2 Experiment design to address research question 3[3]

Equal sized sets are created with document sets with number of authorship known documents varying from one to four. Trained models are tested for these document sets separately and results are recorded and analyzed. Figure 3.5 illustrates the experiment design.

[3] How many authorship known documents are needed to create a successful model of the given author?

3.4.3 Experiment design to address research question 4[4]

For detecting imitation and obfuscation Extended-Brennan-Greenstadt Adversarial Corpus [32] is used which is designed specifically for such tasks in authorship verification. It contains 45 document sets with varying one document with imitation and one document with obfuscation along with other documents.

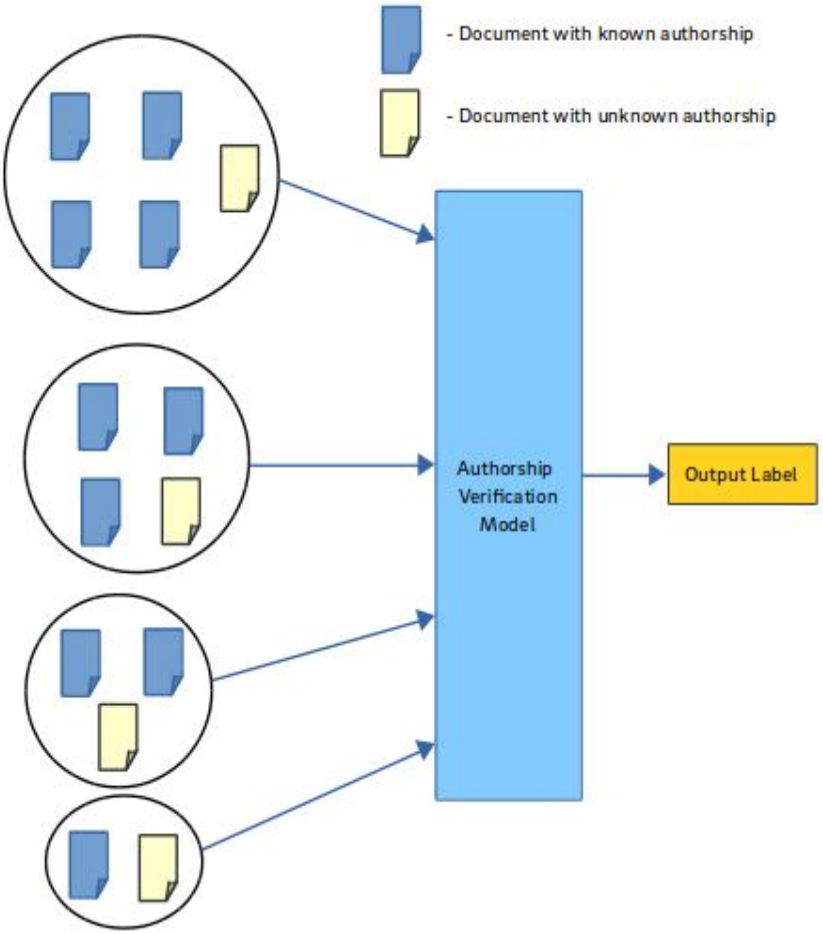

Figure 3.5 - Experiment design to check the effects of variability of processed known authorship documents

3.4.4 Setting up parameters

The ν (nu) parameter, called margin of the one-class SVM is responsible for probability of finding a new but regular data point outside the target margins [34]. The value

[4] How robust the created model would be in case of imitation or obfuscation?

should be in the interval (0, 1]. Hence it needs to be properly setup to distinguish between minute differences between target class and outliers. Experiments are conducted for different values of nu and evaluation matrices are calculated. Figure 3.6 shows the variations of accuracy, recall and precision according different nu values for Model I. Horizontal axis represents various values of nu.

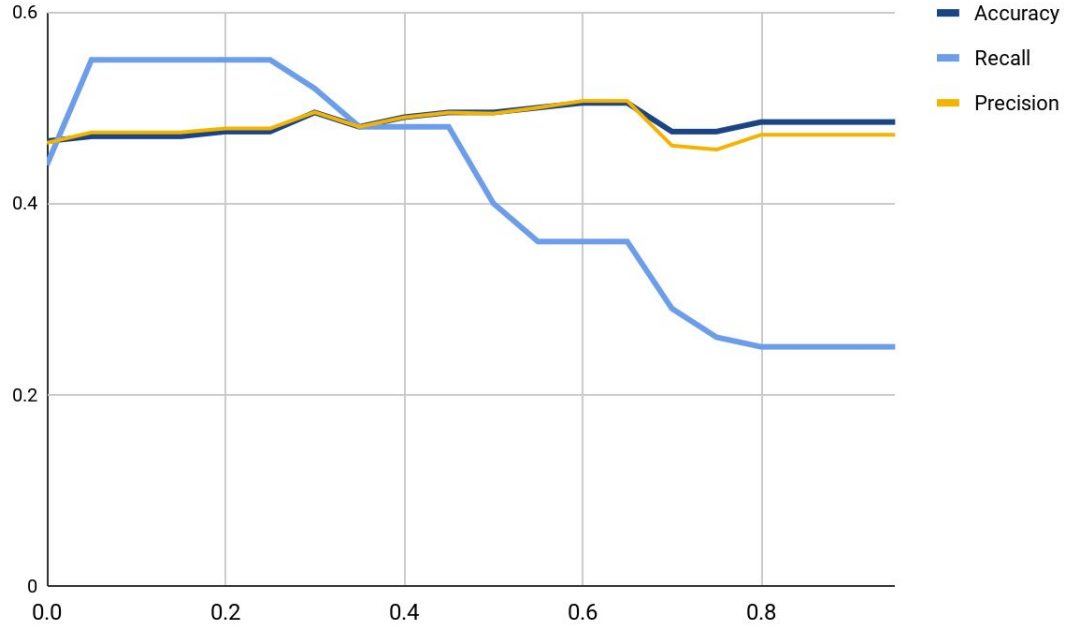

Figure 3.6 - Variations of accuracy, precision and recall according to nu in Model I

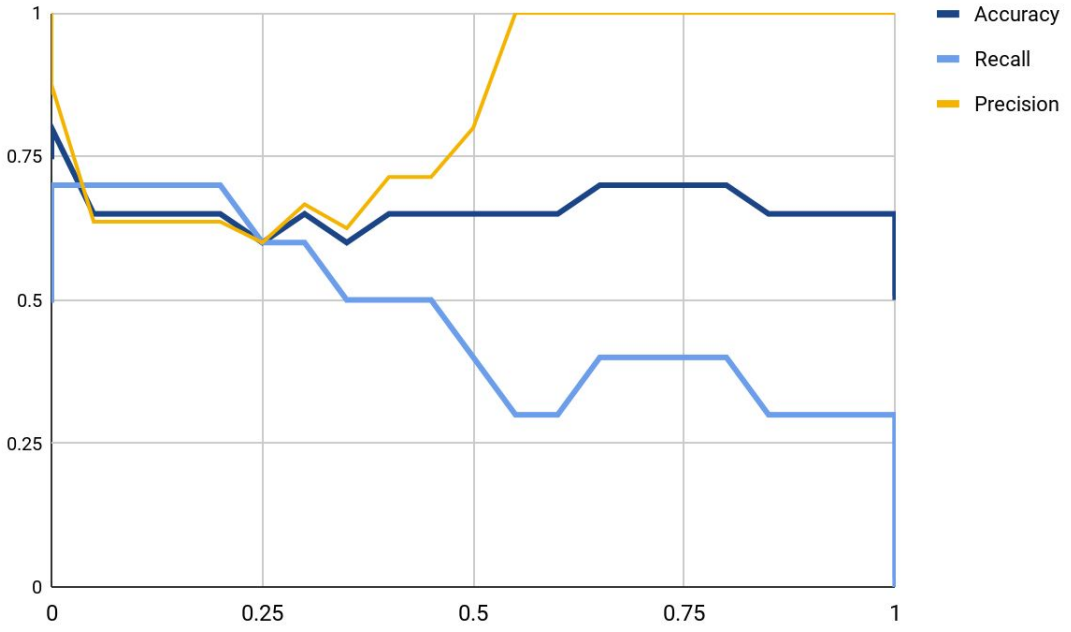

Figure 3.7 - Variations of accuracy, precision and recall according to nu in Model II

After experimentation, optimum values for accuracy, precision and recall are achieved for nu=0.295 for Model I.

Figure 3.7 shows fluctuations of accuracy, precision and recall when value for nu changes in one-class SVM classification. Experiments conducted on various values gave optimum values for accuracy, precision and recall on nu=0.00000051 for Model II.

3.5 Evaluation Design

The original source of performance measures is the coincidence matrix of classification problems [41]. Measures such as True Positive Rate, True Negative Rate, Accuracy, Precision, Recall and F- measure are often used in model evaluation [41]. Formulations of the above measures are shown in Table 3.1.

Table 3.1 - Coincidence Matrix of Performance Measures

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

True Positive Rate is the True Positive Count divided by sum of True Positive Count and False Negative Count (Sum of all positive instances). False Positive Rate is calculated by dividing True Negative Count by sum of False Positive Count and True Negative Count (Sum of all negative instances). Accuracy is the ratio between all the correctly classified instances and all the experimented instances. Precision indicates to what extent the model can retrieve relevant information on rather than irrelevant information, while recall reflects the degree that relevant information is obtained. The improvement on recall can be traded off by lowering precision. Hence F-measure is used to evaluate a model by balancing Precision and Recall. How these measures are calculated is mentioned in Table 3.2.

Table 3.2 - Performance Measures Formulations

Performance Measure	Formulation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
True Positive Rate (Recall)	$\frac{TP}{TP + FN}$
Precision	$\frac{TP}{TP + FP}$
True Negative Rate	$\frac{TN}{TN + FP}$
F1-Measure	$\frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$

Above mentioned performance measures will be used to evaluate the authorship verification models designed above.

When evaluating the models when unknown authorship is being same as known author would be considered as positive class and authorship belonging to a different person than known author would be considered as negative. Hence in cases where authorship is same as known author and model classifies the instance as same then it would be considered as a true positive case. Other measures are calculated in the same way.

Chapter 4 - Implementation

4.1 One-class classification model implementation

According to the research design described in Chapter 3, a classification model is created using python scikit-learn library where one-class SVM with linear kernel is used as the classifier. A linear kernel is selected because most of the text classification problems are linearly separable and linear kernels are faster [48]. In the training phase, features are extracted from known documents of the suspected author and model's one class is trained. Then features are extracted from unknown document and model predicts the specific label ("Y" , "N") of the authorship unknown document. Figure 4.1 shows this process. Same process is carried out for each instance of data set and training accuracies and other evaluation measures are calculated. Similarly testing measures are calculated using testing data.

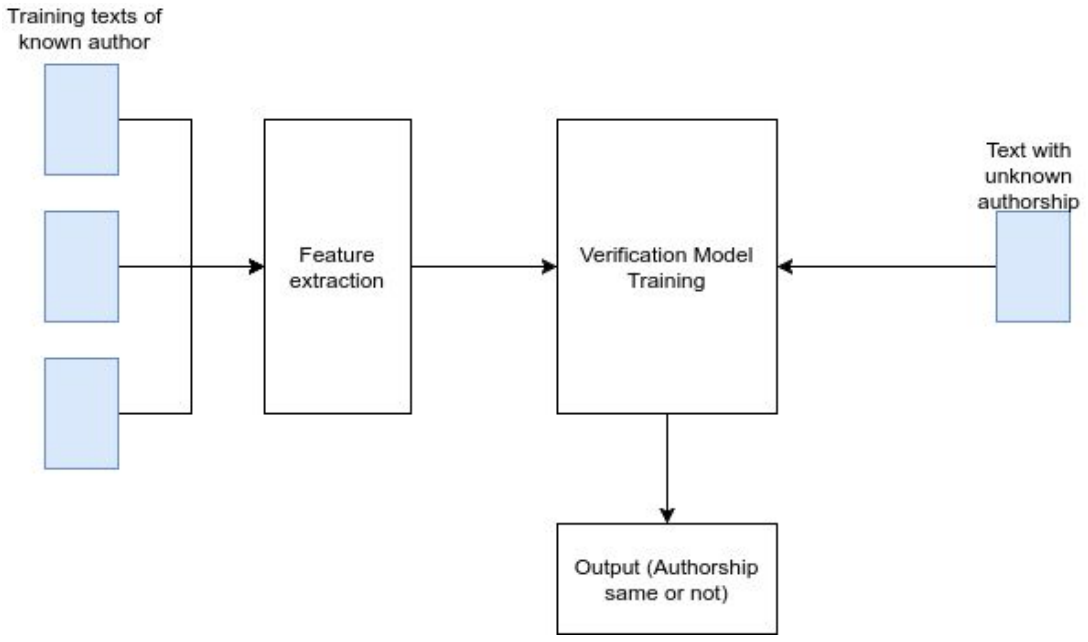

Figure 4.1 - Training of one-class verification model

4.2 Two-class classification model implementation

SVC model in scikit-learn python module, is trained for two classes with extracted features to output “known” or “unknown” labels. Same features described in previous sections are extracted from the known documents. Feature matrix is created in such a way that first few vectors in matrix would be features of the known author and last few vectors would contain features of the unknown author. training labels are assigned in the same order and model is trained as shown in the Figure 4.2. Then model is tested the same way with testing data and evaluation matrices are calculated.

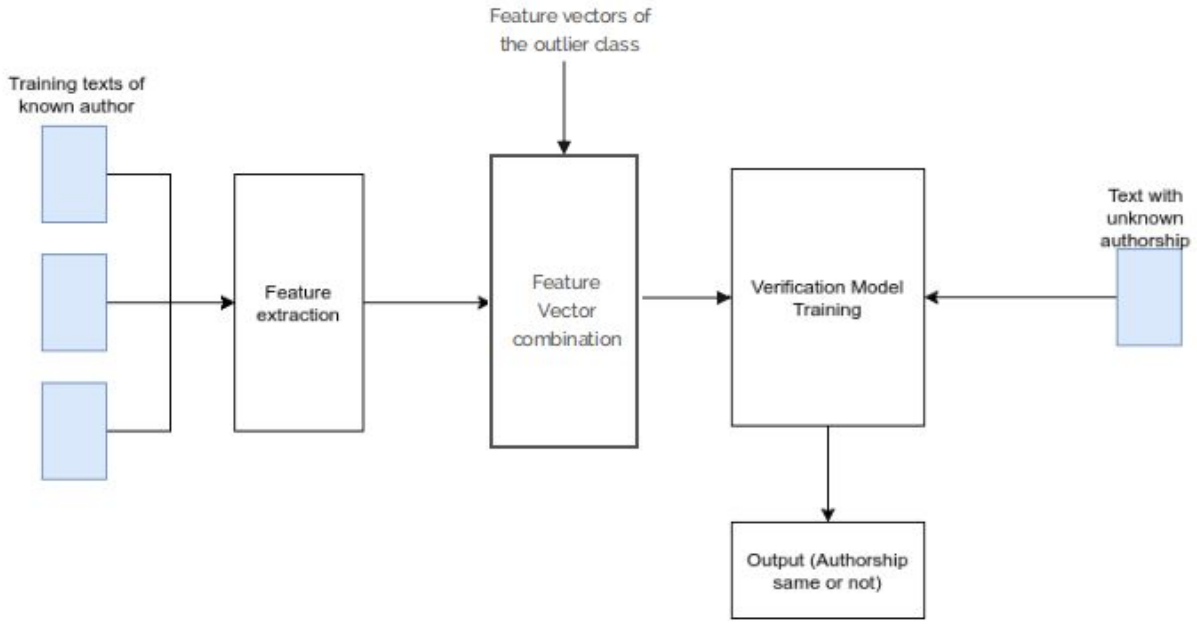

Figure 4.2 - Training of two-class verification model

4.3 Feature extraction - Feature set I

Features selected which are described in section 3.1.1 are extracted using nltk, numpy, stylometry python libraries and are converted to feature vectors and fed to the classification

model created and model is trained and tested. An example of extracted features are shown in Appendix A.

4.4 Feature extraction - Feature set II

Features which are described in feature set II of section 3.1.2 are extracted using the Longman Vocabulary Checker [46]. There are 9000 most frequent words in English identified by Longman Dictionaries and those are divided into three categories as high frequency words, mid frequency words and lower frequency words. The Longman Vocabulary Checker [46] provides an interface to extract information such as percentages of words in those categories when a text is inputted. It also allows to extract word percentage in Academic Word List (AWL) of the given text. Hence the features in feature set II are extracted by providing each known and unknown authorship documents. Feature extraction example is provided in Appendix A. After extracting features feature vectors are created and suitable labels are provided for each feature vector.

Chapter 5 - Results and Evaluation

5.1 Results for experiment for research question 1 and 2

5.1.1 Results on Model I: One-class classification model with Feature set I

The model is trained with 10 authors with different instances and below results were obtained after testing.

When the model is tested with 10 authors in case when each instance, the given unknown document is written by the same author, the model successfully classified all the instances as the given author. When the test set is given with 10 authors with each instance, the given unknown document is written by a different author, the model could identify only one instance as a different author, but all others were identified as the same author, hence giving out more false positives.

Next the model is tested with 10 authors where in 5 instances the unknown document was written by same author and in other 5 instances unknown document is written by a different author. The model successfully classified the instances with unknown document author being same as known documents, but incorrectly classified the instances where unknown document being a different author as the same author. Results are shown in Table 5.1 and 5.2.

The same model is tested with 50 authors selected randomly and tables 5.3 and 5.4 shows the results for different instances.

Table 5.1 - Results from one-class model classification on 10 random instances from the dataset with feature set I

Instance	True Positives	True Negatives	False Positives	False Negatives
All given unknown documents are of the same author	10	0	0	0
All given unknown documents are of the different authors	0	1	9	0
Given unknown document being same and different authors randomly	5	0	5	0

Table 5.2 - Performance measures from one-class model classification on 10 random instances from the dataset with feature set I

Instance	Accuracy	Recall	Precision	F1
All given unknown documents are of the same author	1.0000	1.0000	1.0000	1.0000
All given unknown documents are of the different authors	0.1000	-	0.0000	-
Given unknown document being same and different authors randomly	0.5000	1.0000	0.5000	0.6667

Table 5.3 - Results from one-class model classification on 50 random instances from the dataset with Feature set I

True Positives	True Negatives	False Positives	False Negatives
24	2	23	1

Table 5.4 - Performance measures from one-class model classification on 50 random instances from the dataset with Feature set I

Accuracy	Recall	Precision	F1
0.5200	0.9600	0.5106	0.6667

When model is tested with document instances with unknown document being written by same author and different author randomly (where both instances appear) it correctly classified instances where unknown document written by same author as known documents where 24 out of 25. But when the unknown document is written by a different author, the classifier classified them as same author in 23 instances out of 25. Hence giving an accuracy of 0.52 and recall of 0.96 and precision of 0.5106.

The same model is tested with 200 instances of PAN test data and results are shown in table 5.5 and table 5.6. It shows that one-class classification model with feature set I performs poorly in identifying same author cases as well as different author cases.

Table 5.5 - Results from one-class model classification with feature set I on 200 cases

True Positives	True Negatives	False Positives	False Negatives
55	39	61	45

Table 5.6 - Performance measures from one-class model classification with feature set I on 200 cases

Accuracy	Recall	Precision	F1
0.4700	0.5500	0.4741	0.5093

When analyzing the results of above experiments the model does not show any significant improvement over the variability of number of authors introduced (when number of authors increase from 10 to 50). The model has shown accuracies 0.5, 0.52 and 0.47 with 10 authors, 50 authors and 200 authors. In 10 author case and 50 author case a higher recall is

shown. This indicates the model tends to classify document instances with label "Y" indicating the document is written by the same author as the given author, but does not perform well, when the unknown document is actually written by a different author. 0.55 (true positive rate) recall and 0.39 of false positive rate is achieved on total data set of 200 instances of authors. So when the model is tested on a larger test set it shows that model performs poorly in both cases but more improvement in identifying same author cases.

5.1.2 Results on Model II: One-class classification model with Feature set II

The model is tested with 10 document sets obtained from sample of 50 to check the model behaviour of each instance mentioned in section 3.4.1.1. Results are shown in table 5.7 and table 5.8.

Table 5.7 - Results from one-class model classification on 10 random instances from the dataset with feature set II

Instance	True Positives	True Negatives	False Positives	False Negatives
All given unknown documents are of the same author	7	0	0	3
All given unknown documents are of the different authors	0	9	1	0
Given unknown document being same and different authors randomly	3	4	1	2

Table 5.8 - Performance measures from one-class model classification on 10 random instances from the dataset with feature set II

Instance	Accuracy	Recall	Precision	F1
All given unknown documents are of the same author	0.7000	0.7000	1.0000	0.8235
All given unknown documents are of the different authors	0.9000	-	0.0000	-
Given unknown document being same and different authors randomly	0.7000	0.6000	0.7500	0.6667

When one-class model with feature set II is tested on 10 author instances, when all the unknown document is same as given author the model has classified 7 correctly and 3 incorrectly as different author. In this case model has performed well. When document sets with unknown document of a different author than suspected author are given, the classification shows 9 as different author and just one instance as same author, hence giving an accuracy of 0.9. In case of mixed instances with both unknown document belonging to same author as well as unknown document belonging to a different author, out of 5 same cases 3 are classified as same and 2 as different and out of 5 different cases 4 are classified as same and just one is classified incorrectly. Hence the model shows potential in identifying the authorship when the unknown document is of same author as well as of a different author.

Next the experiments were continued to check the behaviour of model when number of authors increase. Hence after training and testing the one-class SVM with sample datasets, randomly selected of 50 document sets with 50 authors, below results were obtained as shown in table 5.9 and 5.10. The sample dataset contained 25 instances with unknown author being the same author as given author and 25 instances with unknown author being a different author.

Table 5.9 - Results from one-class SVM on testing dataset of 50 with feature set II

True Positives	True Negatives	False Positives	False Negatives
18	22	3	7

Table 5.10 - Performance measures from one-class SVM on testing dataset of 50 with feature set II

Accuracy	Recall	Precision	F1
0.8000	0.7200	0.8751	0.7900

From the results in Table 5.9 and 5.10 it shows that the one-class classification model has performed well on the feature set II. The model also shows well balanced classification on both positive and negative classes unlike the classification of the model using feature set I.

5.1.3 Results on Model III: Two-class classification model with Feature set I

After training the model, the two-class classification model is first tested with 10 instances of three cases which are mentioned in section 3.4.1.1. From the results in table 5.11 and table 5.12 it shows that, when only unknown documents with same author is given the model has performed better giving an accuracy of 0.9. But when cases with unknown document of different authors given the model could identify only one case and failed in all other cases. When the model is tested with mixed cases of unknown being same and unknown being different again it shows tendency to output label, “Y” correctly identifying only four same author cases and one different author cases. It shows that model will perform poorly when unknown document is of a different author.

Table 5.11 - Results from two-class model classification on 10 random instances from the dataset with feature set I

Instance	True Positives	True Negatives	False Positives	False Negatives
All given unknown documents are of the same author	9	0	0	1
All given unknown documents are of the different authors	0	1	9	0
Given unknown document being same and different authors randomly	4	1	4	1

Table 5.12 - Performance measures from two-class model classification on 10 random instances from the dataset with feature set I

Instance	Accuracy	Recall	Precision	F1
All given unknown documents are of the same author	0.9000	0.0000	0.0000	-
All given unknown documents are of the different authors	0.100	-	1.0000	-
Given unknown document being same and different authors randomly	0.5000	0.8000	0.5000	0.6154

Next, the two class classification model is evaluated with top 50 authors from Reuters Corpus Volume 1 [39, 40]. Training and testing datasets are given separately and each contains 50 instances of documents for each author. Dataset is changed according to research needs where in random 25 authors the document which is unknown is obtained from a different author of the same dataset. Hence there are 25 instances with unknown author being

same and 25 instances with unknown author being different. Experiment is conducted on the dataset as described in section 4.2. Table 5.14 shows the evaluation matrices obtained.

Table 5.13 - Results from two-class model classification on 50 authors - feature set I

True Positives	True Negatives	False Positives	False Negatives
24	7	18	1

Table 5.14 - Performance measures from two-class model classification on 50 authors with feature set I

Accuracy	Recall	Precision	F1
0.6200	0.7200	0.4286	0.5373

The results show that even though the model can successfully identify the instances where unknown author is same as the known author other instances cannot be successfully distinguished. This however, can be improved if the outlier class of the model be more properly represented.

The same model is evaluated with 200 cases of PAN corpora. Results are shown in table 5.15 and table 5.16. According to obtained results the two-class classification model has classified same author cases successfully while the different author cases are poorly classified.

Table 5.15 - Results from two-class model classification with feature set I on 200 cases

True Positives	True Negatives	False Positives	False Negatives
89	14	86	11

Table 5.16 - Performance measures from two-class model classification with feature set I on 200 cases

Accuracy	Recall	Precision	F1
0.5150	0.8600	0.4914	0.6255

5.2 Discussion for experiments on research question 1 and 2

According to comparisons in table 5.17 on candidate models, Model II shows the highest accuracy, precision and F1-score, while Model I shows the highest recall. But Model I or Model III does not show a balance between precision and recall while Model II does. This indicates Model II is capable to identify the suspected author's writing style as well as differentiate if the unknown document is of a different author. Both Model I and Model II tends to identify same author cases but fails to identify different author cases. Hence Model II outperforms other models regarding research questions 1 and 2.

Table 5.17 - Performance measures comparison for candidate models

Performance Measure	Model I	Model II	Model III
Accuracy	0.5200	**0.8000**	0.6200
Recall	**0.9600**	0.7200	0.7200
Precision	0.5106	**0.8751**	0.4286
F1-Score	0.6667	**0.7900**	0.5373

5.3 Results for experiment for research question 3

The experiment is conducted all three models designed in section 3.2. intended to check the performance of verification when number of known documents provided varies. Hence 10 document sets are created for each instance where number of known documents varies from one to four.

5.3.1 Results on Model I: One-class classification model with Feature set I

First the on one-class classification model with feature set I, designed in section 3.2.1. are subjected to the experiment. Results are shown in table 5.18 and 5.19.

Table 5.18 - Results obtained when number of given known documents varies on Model I

Number of known documents	True Positives	True Negatives	False Positives	False Negatives
1	1	2	3	4
2	2	3	3	2
3	4	1	5	0
4	4	0	5	1

Table 5.19 - Performance measures obtained when number of given known documents varies on Model I

Number of known documents	Accuracy	Recall	Precision	F1
1	0.3000	0.2000	0.2500	0.2222
2	0.5000	0.5000	0.4000	0.4444
3	0.5000	1.0000	0.4444	0.6154
4	0.4000	0.8000	0.4444	0.5714

According to table 5.19, when the number of known documents given is one it gives the lowest accuracy and F1 score. This is because the given feature vector carries less data to train the model to fit a sufficient representation of the author's style. But accuracy and F1 score tends to increase when number of documents with known authorship increase since it contributes to create a sufficient profile for the author. But when number of known documents is four the accuracy and F1 has decreased, this could be because the selected sample does not properly represent the author's style.

5.3.2 Results on Model II: One-class classification model with Feature set II

One-class classification model with feature II is subjected to experimentation to see the model behaviour in case of various number of known documents are introduced. Results are shown in table 5.20 and 5.21.

Table 5.20 - Results obtained when number of given known documents varies on Model II

Number of known documents	True Positives	True Negatives	False Positives	False Negatives
1	2	4	1	3
2	3	4	1	2
3	4	3	2	1
4	4	3	2	1

Table 5.21 - Performance measures obtained when number of given known documents varies on Model II

Number of known documents	Accuracy	Recall	Precision	F1
1	0.6000	0.4000	0.6667	0.5000
2	0.7000	0.7500	0.6000	0.6667
3	0.7000	0.6667	0.8000	0.7273
4	0.7000	0.6667	0.8000	0.7273

According to results in table 5.21, when number of known documents increases from one to two all the evaluation matrices has increased. When known documents number is three and four accuracies and other measures stay the same. At this point F1-score has increased which shows the balance between precision and recall. Hence there should be at least two documents for a better classification in Model II.

5.3.3 Results on Model III: Two-class classification model with Feature set I

Two-class classification model is subjected to tests to check how the model would behave in case of variability of number of documents provided for a known author. Results are in table 2.22 and 2.23.

Table 5.22 - Results obtained when number of given known documents varies on Model III

Number of known documents	True Positives	True Negatives	False Positives	False Negatives
1	3	1	4	2
2	3	1	5	1
3	4	0	6	0
4	5	0	5	0

Table 5.23 - Performance measures obtained when number of given known documents varies on Model III

Number of known documents	Accuracy	Recall	Precision	F1
1	0.4000	0.8000	0.5714	0.6667
2	0.4000	0.2500	0.6250	0.8333
3	0.4000	0.5000	0.6000	0.8571
4	0.5000	1.0000	0.5000	0.6667

According to results in table 5.22 and 5.23 is can be realized that when the number of documents used for a known author increases, the accuracy has increased. Also recall has increased dramatically and F1 score gets reasonable value. Hence Model II shows an improvement in classification with high number of known documents which is at least four.

5.3 Results for experiment for research question 4

This experiment is conducted to observe the model behaviour in case of imitation or obfuscation. It is conducted on one-class classification model with feature set I designed in section 3.2.1. Documents of 10 authors are selected for this experiment, with each unknown authorship document being an attempt to imitate the writing style of the author. However the results in table 5.24 and 5.25 shows that verification model does not perform well in case of imitation where it has only identified one case of ten as an imitation.

Table 5.24 - Results from model classification on 10 authors with imitation

True Positives	True Negatives	False Positives	False Negatives
0	1	9	0

Table 5.25 - Performance measures from model classification on 10 authors with imitation

Accuracy	Recall	Precision	F1
0.1000	-	0.0000	-

Chapter 6 - Conclusions

6.1 Introduction

This research attempted on creating a novel approach for authorship verification which adopts supervised learning technique. Two feature sets are developed and combined with two models. Hence three different models are designed and evaluated:

1. Model I: One-class classification model with Feature set I
2. Model II: One-class classification model with Feature set II
3. Model III: Two-class classification model with Feature set I

According to results obtained in the experiments it is shown that one-class classification model with feature set II performs better.

6.2 Conclusions about research questions

1. Will the author be verified if document with unknown authorship is same as of documents with known authorship ?

Regarding the findings for Model I, it shows that feature set I captures the writing style of an author, but fails to distinguish if a different writing style is introduced from the given writing style. Hence Model I will successfully output correct label if a document set with unknown document being same is given. Model II also performs well in this aspect but not as well as Model I, indicating the frequency percentages of word lists are able to identify the writing style of a person but not as much as stylometric features. Model III also shows similar performance indicating SVM with linear kernel can identify target class when model is trained with data of a suspected author. Hence all three models performs better in solving research question 1 but Model I outperforms others.

2. Will the author be not verified if document with unknown authorship is not same as of documents with known authorship ?

In Model I, true negative rate of 0.52 is achieved and two-class classification model gets 0.28 true negative rate. Hence both models show low ability to correctly classify instances where unknown document is written by a different author than the suspected author. This shows that even though stylometric features in feature set I are capable of identifying a writing style, they fail to distinguish between different writing styles when difference is minute. Two-class classification also shows lower performance in identifying such cases than one-class classification model. This could be because of the poor representation of the outlier class. But one-class classification model with feature set II indicated a 0.88 of true negative rate showing the highest performance in identifying negative instances.

3. How many documents with author known to create a successful model of the given author?

Performance measures have been increased when the number of known documents for a certain author increases in all three models. This could be because when the number of known documents increase there are more data which represent the writing style of the author. Experiments showed that Model I and Model III needed at least four documents for better classification while Model II only needed two documents at least to achieve a better classification. Hence it can be concluded that Model II is capable of distinguishing writing styles of authors even with the presence of less number of data than other models.

4. How robust the created model would be in case of imitation or obfuscation?

In case of imitation or obfuscation the one-class classification model has shown a very low accuracy of 0.1. Hence it can be concluded that model does not perform well in case of imitation or obfuscation of writing style of the given unknown document.

6.3 Conclusions about research problem

According to results and conclusions gained for each research question in sections 5 and 6, it can be observed that the developed one-class classification model with feature set I, performs very poorly and predictions obtained from the model are not reliable. Two-class classification gives accuracy of 0.62 but in case of identifying unknown author being the same author as the suspected author, model shows potential since it identified 24 cases out of 25. Model performs poorly when unknown document is of a different author than the known author. In these cases, the model behaviour can be improved by choosing a good representation for the outlier class. However when one-class classification model is trained and tested utilizing feature set II, it shows the highest accuracy of the three models as 0.8. Hence Model II better performs in solving the research problem.

From one-class classification model with feature set I, it was attempted to test if selected 21 features suggested in Hanlein's empirical research [28, 29] to characterize the writing style of a person can be successfully employed in one-class SVM to solve the authorship verification problem. But results suggest otherwise. Hence it can be concluded that one-class SVM is unable to distinguish between different writing styles based on the features extracted in feature set I.

When one-class classification model is implemented with feature set II, it shows high potential in tackling the authorship verification problem, indicating the features in selected feature set can successfully distinguish between different writing styles. This feature set is concerned about an author's nature of vocabulary usage and its richness, indicating different authors use different vocabulary levels and they remain same across multiple documents of the same author over a certain period of time.

Two class classification model is designed to use same features as feature set I and SVM trained for two classes, known author and unknown author. This shows potential in

distinguishing between different writing styles of authors when the outlier class is properly represented.

6.4 Limitations

The models designed and implemented in this research, most likely will not be able to be applied in most of the real-life scenarios of authorship verification. Since most of the documents, disputed in authorship could be very short such as emails, online messages, ransom notes etc. these methods which are applied for 1000 word documents will not be successful for such cases.

In case of two-class classification models, creating a good representation for the outlier class is very challenging and the results are solely based on the improvement of the outlier class. In such cases the two-class classification model becomes highly language dependent. Also the outlier class needs to be closer to the target class and this is another constraint which needs to be satisfied and it is highly challenging.

6.5 Implications for further research

In case of two-class classification model the outlier class representation can be improved to achieve better performance. A better approach needs to be proposed to create a outlier class representation which is closer to the target class.

One-class classification model with feature set II can be further tested increasing number of document instances and number of authors utilized observing the model behaviour in case of a larger dataset.

Behaviours of Model II and Model III can be further tested regarding the research question 4. It would be interesting to see the model behaviours in case of unknown author tries to imitate or obfuscate the known author.

References

[1] R. Zheng, J. Li, H. Chen and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques", Journal of the American Society for Information Science and Technology, vol. 57, no. 3, pp. 378-393, 2006.

[2] M. Koppel, J. Schler and S. Argamon, "Computational methods in authorship attribution", Journal of the American Society for Information Science and Technology, vol. 60, no. 1, pp. 9-26, 2009.

[3] P. Juola, "Authorship Attribution", Foundations and Trends® in Information Retrieval, vol. 1, no. 3, pp. 233-334, 2007.

[4] F. Mosteller and D. Wallace, "Inference in an Authorship Problem", Journal of the American Statistical Association, vol. 58, no. 302, pp. 275-309, 1963.

[5] "Breaking News, Analysis, Politics, Blogs, News Photos, Video, Tech Reviews - TIME.com",TIME.com,2009.[Online].Available:http://content.time.com/time/arts/article/0,8599,1930971,00.html. [Accessed: 11- Apr- 2017].

[6] K. Rasheed, C. He and Ramyaa, "Using Machine Learning Techniques for Stylometry", Proceedings of the International Conference on Artificial Intelligence, vol. 2, no. -04, 2004.

[7] Z. Li, "An Exploratory Study on Authorship Verification Models for Forensic Purpose", Master of Science, Knowledge and Expertise Center for Intelligent Data Analysis (KECIDA), Netherlands Forensic Institute, 2013.

[8] S. Nirkhi, R. Dharaskar and V. Thakare, "Authorship Verification of Online Messages for Forensic Investigation", Procedia Computer Science, vol. 78, pp. 640-645, 2016.

[9] F. Iqbal, L. Khan, B. Fung and M. Debbabi, "e-mail authorship verification for forensic investigation", Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10, 2010.

[10] A. Stolerman, "Authorship Verification", Ph.D, Drexel University, 2015.

[11] K. Luyckx and W. Daelemans, "Authorship Attribution and Verification with Many Authors and Limited Data", Proceeding COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 513-520, 2008.

[12] H. Baayen, H. van Halteren and F. Tweedie, "Outside the cave of shadows: using syntactic annotation to enhance authorship attribution", Literary and Linguistic Computing, vol. 11, no. 3, pp. 121-132, 1996.

[13] E. Stamatatos, "A survey of modern authorship attribution methods", Journal of the American Society for Information Science and Technology, vol. 60, no. 3, pp. 538-556, 2009.

[14] M. Koppel and J. Schler, "Authorship Verification as a One-Class Classification Problem", The 21st International Conference on Machine Learning. (ICML-04), 2004.

[15] S. Argamon, C. Whitelaw, P. Chase, S. Hota, N. Garg and S. Levitan, "Stylistic text classification using functional lexical features", Journal of the American Society for Information Science and Technology, vol. 58, no. 6, pp. 802-822, 2007.

[16] D. M. J. Tax, "One-class classification; Concept-learning in the absence of counter-examples", Ph.D, Delft University of Technology, 2001.

[17] J. Noecker Jr and M. Ryan, "Distractorless Authorship Verification", Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), pp. 785-789, 2012.

[18] A. Glover and G. Hirst, "Detecting Stylistic Inconsistencies in Collaborative Writing", The New Writing Environment, pp. 147-168, 1996.

[19] J. Grieve, "Quantitative Authorship Attribution: An Evaluation of Techniques", Literary and Linguistic Computing, vol. 22, no. 3, pp. 251-270, 2007.

[20] M. Corney, "Analysing E-mail Text Authorship for Forensic Purpose ", Master thesis, University of Software Engineering and Data Communications, 2003.

[21] M. Koppel, J. Schler, S. Argamon and E. Messeri, "Authorship attribution with thousands of candidate authors", Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '06, 2006.

[22] G. Hirst and O. Feiguina, "Bigrams of Syntactic Labels for Authorship Discrimination of Short Texts", Literary and Linguistic Computing, vol. 22, no. 4, pp. 405-417, 2007.

[23] J. Burrows, "'Delta': a Measure of Stylistic Difference and a Guide to Likely Authorship", Literary and Linguistic Computing, vol. 17, no. 3, pp. 267-287, 2002.

[24] D. Hoover, "Testing Burrows's Delta", Literary and Linguistic Computing, vol. 19, no. 4, pp. 453-475, 2004.

[25] D. Hoover, "Delta Prime?", Literary and Linguistic Computing, vol. 19, no. 4, pp. 477-495, 2004.

[26] M. Brocardo, I. Traore, S. Saad and I. Woungang, "Authorship verification for short messages using stylometry", 2013 International Conference on Computer, Information and Telecommunication Systems (CITS), 2013.

[27] S. Mechti, M. Jaoua, R. Faiz and L. Hadrich Belguith, "An Analysis Framework for Hybrid Authorship Verification", Research in Computing Science, vol. 110, pp. 151-158, 2016.

[28] K. Rasheed, C. He and Ramyaa, "Using Machine Learning Techniques for Stylometry", Proceedings of the International Conference on Artificial Intelligence, vol. 2, no. -04, 2004.

[29] Hanlein, H. “Studies in Authorship Recognition: a Corpus-based Approach”. Peter Lang, 1999.

[30] "PAN", Pan.webis.de, 2017. [Online]. Available: http://pan.webis.de/data.html. [Accessed: 12- Mar- 2017].

[31] Halvani, Oren (2016), “Reddit Cross-Topic Authorship Verification Corpus”, Mendeley Data, v1

[32] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. ACM Trans. Inf. Syst. Secur. 15, 3, Article 12 (November 2012).

[33] "Unabomber", Federal Bureau of Investigation, 2017. [Online]. Available: https://www.fbi.gov/history/famous-cases/unabomber. [Accessed: 13- May- 2017].

[34] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola and R. Williamson, "Estimating the Support of a High-Dimensional Distribution", Neural Computation, vol. 13, no. 7, pp. 1443-1471, 2001.

[35] J. Burrows, "Word-Patterns and Story-Shapes: The Statistical Analysis of Narrative Style", Literary and Linguistic Computing, vol. 2, no. 2, pp. 61-70, 1987.

[36] R. Forsyth and D. Holmes, "Feature-Finding for Text Classification", Literary and Linguistic Computing, vol. 11, no. 4, pp. 163-174, 1996.

[37] F. Iqbal, R. Hadjidj, B. Fung and M. Debbabi, "A novel approach of mining write-prints for authorship attribution in e-mail forensics", Digital Investigation, vol. 5, pp. S42-S51, 2008.

[38] F. Tweedie, S. Singh and D. Holmes, "Neural network applications in stylometry: The Federalist Papers", Computers and the Humanities, vol. 30, no. 1, pp. 1-10, 1996.

[39] D. Khmelev and W. Teahan, "A repetition based measure for verification of text collections and for text categorization", Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '03, 2003.

[40] D. Lewis, Y. Yang, T. Rose and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research", Journal of Machine Learning Research, vol. 5, pp. 361-397, 2004.

[41] D. Olson and D. Delen, Advanced data mining techniques. Berlin: Springer, 2008.

[42] S. Eissen and B. Stein, "Intrinsic Plagiarism Detection", Lecture Notes in Computer Science, pp. 565-569, 2006.

[43] I. Nation, "How Large a Vocabulary Is Needed for Reading and Listening?", The Canadian Modern Language Review / La revue canadienne des langues vivantes, vol. 63, no. 1, pp. 59-81, 2006.

[44] P. Nation, "How much input do you need to learn the most frequent 9,000 words?", Reading in a Foreign Language, vol. 26, no. 2, pp. 1-16, 2014.

[45] A. Chi, "A review of Longman Dictionary of Contemporary English (6th edition)", Lexicography, vol. 2, no. 2, pp. 179-186, 2016.

[46] "Longman Vocabulary Checker", Longmandictionariesusa.com, 2017. [Online]. Available: http://www.longmandictionariesusa.com/vocabulary_checker. [Accessed: 14- Nov- 2017].

[47] A. Coxhead, "A New Academic Word List", TESOL Quarterly, vol. 34, no. 2, p. 213, 2000.

[48] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features", Machine Learning: ECML-98, pp. 137-142, 1998.

Appendix A: Diagrams

A.1 Example: Feature extraction employing feature set I

```
Name:  1213471newsML.txt

>>> Phraseology Analysis <<<

Lexical diversity         : 50.578338591
Mean Word Length          : 5.74012474012
Mean Sentence Length      : 23.6285714286
STDEV Sentence Length     : 12.2126064008
Mean paragraph Length     : 827.0
Document Length           : 5355

>>> Punctuation Analysis (per 1000 tokens) <<<

Commas                    : 42.0609884332
Semicolons                : 5.25762355415
Quotations                : 0.0
Exclamations              : 0.0
Colons                    : 1.05152471083
Hyphens                   : 0.0
Double Hyphens            : 0.0

>>> Lexical Usage Analysis (per 1000 tokens) <<<

and                       : 9.46372239748
but                       : 5.25762355415
however                   : 0.0
if                        : 0.0
that                      : 4.20609884332
more                      : 1.05152471083
must                      : 0.0
might                     : 0.0
this                      : 7.36067297581
very                      : 1.05152471083

################################################
Name:  136471newsML.txt

>>> Phraseology Analysis <<<

Lexical diversity         : 55.900621118
Mean Word Length          : 5.51666666667
Mean Sentence Length      : 24.1666666667
STDEV Sentence Length     : 10.9379563397
Mean paragraph Length     : 290.0
Document Length           : 1723

>>> Punctuation Analysis (per 1000 tokens) <<<

Commas                    : 40.3726708075
Semicolons                : 0.0
Quotations                : 0.0
Exclamations              : 0.0
Colons                    : 0.0
Hyphens                   : 0.0
Double Hyphens            : 9.31677018634

>>> Lexical Usage Analysis (per 1000 tokens) <<<

and                       : 21.7391304348
but                       : 3.10559006211
however                   : 3.10559006211
if                        : 0.0
that                      : 3.10559006211
more                      : 3.10559006211
must                      : 3.10559006211
```

Figure A.1 - Feature extraction from documents - Feature set I


{"out": [["EN04", [[[46.32911392405064, 6.6502732240437155, 17.566666666666666, 9.360140074925283, 1054.0, 6986, 45.56962025316456, 0.0, 0.0, 0.0, 2.5316455696202533,
22.78481012658228, 1.6877637130801688, 0.0, 0.0, 4.219409282700422, 2.5316455696202533, 0.0, 0.8438818565400844, 3.375527426160337, 1.6877637130801688], [42.07699194270367,
6.729787234042553, 15.166666666666666, 9.820484671945636, 1001.0, 6665, 35.810205908683976, 0.0, 0.0, 0.0, 0.0, 30.43867502238138, 0.0, 0.0, 3.5810205908683974, 7.1620411817736795,
2.6857654431512983, 0.8952551477170994, 0.0, 9.847806624888094, 0.0], [39.621087314662276, 6.667359667359667, 17.508196721311474, 6.9154684082542826, 1068.0, 7226, 32.948929159802304,
0.0, 0.0, 0.0, 0.0, 28.006589785831960, 1.6474464579901154, 0.0, 0.8237232289950577, 8.237232289950576, 2.471169686985173, 0.0, 0.0, 1.6474464579901154, 0.8237232289950577],
[44.62051709758132, 6.906542056074766, 15.098591549295774, 8.028391397957739, 1072.0, 7234, 36.69724770642202, 0.0, 0.0, 0.0, 0.8340283569641368, 38.36530442035029, 0.0, 0.0, 0.0,
6.672226855713094, 2.5020850708924103, 0.8340283569641368, 0.0, 5.004170141784821, 1.6680567139282736], [38.09924306139613, 6.335540838852097, 17.306451612903224, 9.607773582646534,
1073.0, 6751, 26.07232968881413, 0.0, 0.0, 0.0, 0.0, 16.82085786375105, 1.6820857863751050, 0.0, 0.8410428931875525, 8.410428931875526, 0.0, 2.5231286795626575, 1.6820857863751050,
5.887300252312868, 0.8410428931875525]], ["same", "same", "same", "same", "same"]]], ["EN07", [[[46.32911392405064, 6.6502732240437155, 17.566666666666666, 9.360140074925283, 1054.0,
6986, 45.56962025316456, 0.0, 0.0, 0.0, 2.5316455696202533, 22.78481012658228, 1.6877637130801688, 0.0, 0.0, 4.219409282700422, 2.5316455696202533, 0.0, 0.8438818565400844,
3.3755274261603376, 1.6877637130801688], [42.07699194270367, 6.729787234042553, 15.166666666666666, 9.820484671945636, 1001.0, 6665, 35.810205908683976, 0.0, 0.0, 0.0, 0.0,
30.43867502238138, 0.0, 0.0, 3.5810205908683974, 7.162041181736795, 2.6857654431512983, 0.8952551477170994, 0.0, 9.847806624888094, 0.0], [39.621087314662276, 6.667359667359667,
17.508196721311474, 6.9154684082542826, 1068.0, 7226, 32.948929159802304, 0.0, 0.0, 0.0, 0.0, 28.006589785831960, 1.6474464579901154, 0.0, 0.8237232289950577, 8.237232289950576,
2.471169686985173, 0.0, 0.0, 1.6474464579901154, 0.8237232289950577], [44.62051709758132, 6.906542056074766, 15.098591549295774, 8.028391397957739, 1072.0, 7234, 36.69724770642202,
0.0, 0.0, 0.0, 0.8340283569641368, 38.36530442035029, 0.0, 0.0, 0.0, 6.672226855713094, 2.5020850708924103, 0.8340283569641368, 0.0, 5.004170141784821, 1.6680567139282736],
[36.84210526315789, 6.711904761904762, 23.477272727272727, 9.473620399354957, 1033.0, 6514, 39.473684210526315, 0.0, 0.0, 0.0, 0.0, 39.473684210526315, 1.7543859649122806, 0.0,
2.63157894736842112, 6.140350877192983, 0.8771929824561403, 1.7543859649122806, 0.0, 0.0, 0.8771929824561403]], ["same", "same", "same", "same", "not"]]], ["EN11",
[[[32.67489711934156, 6.10831234256927, 20.877551020408163, 10.799910530735279, 1023.0, 6263, 70.78189300411523, 0.823045267489712, 0.0, 0.0, 0.0, 24.691358024691358, 0.0, 0.0,
0.823045267489712, 1.646090534979424, 0.0, 0.823045267489712, 0.0, 0.0, 0.0, 0.0], [27.166666666666668, 5.8619631901840494, 24.595238095238095, 19.367448788169245, 1032.0, 5965,
50.833333333333336, 0.0, 0.0, 0.0, 0.0, 0.0, 17.5, 0.0, 0.0, 2.5, 2.5, 0.0, 0.0, 0.0, 0.0, 1.6666666666666667, 0.0], [29.501267962806427, 5.974212034383954, 19.269230769230770,
11.927852641577188, 1000.0, 6067, 43.11073541842772, 0.0, 0.0, 0.0, 0.0, 26.204564666103128, 0.0, 0.0, 0.0, 0.8453085376162299, 0.8453085376162299, 0.0, 0.8453085376162299, 0.0,
1.69061707523245599, 0.8453085376162299]], ["same", "same", "same"]]], ["EN13", [[[31.180017226528854, 5.5220994475138125, 17.120689655172413, 8.646461477165088, 993.0, 5612,
56.84754521963824, 0.8613264427217916, 0.0, 0.8613264427217916, 0.0, 21.533161068044787, 0.0, 0.0, 0.0, 8.613264427217915, 0.8613264427217916, 0.0, 0.0, 6.890611541774333,
0.8613264427217916], [26.356589147286826, 5.885620915032680, 21.04, 8.722293276426791, 1052.0, 5871, 22.394487510766658, 1.7226528854435832, 0.0, 0.0, 0.0, 20.671834625322997,
0.8613264427217916, 0.0, 0.0, 16.36520241171404, 2.5839793281653747, 0.0, 0.0, 3.4453057708871664, 0.0], [30.21834061135371, 5.713872832369942, 20.857142857142858, 11.408911661772233,
1022.0, 5862, 34.93449781659389, 0.0, 0.0, 0.0, 0.0, 20.960698680995633, 3.493449781659389, 0.0, 2.6200873362445414, 11.353711790393014, 3.493449781659389, 0.0, 0.0,
2.6200873362445414, 0.8733624454148472], [29.604709840201855, 5.605113636363637, 19.26923076923077, 12.019399998808632, 1000.0, 5576, 40.37005887300252, 0.8410428931875525, 0.0, 0.0,
0.0, 16.820857863751050, 1.6820857863751050, 0.0, 1.6820857863751050, 2.5231286795626575, 0.0, 0.0, 0.0, 0.8410428931875525, 0.0]], ["same", "same", "same", "not"]]], ["EN18",
[[[34.07344150299889, 5.807017543859649, 17.633333333333333, 9.470949735316351, 1058.0, 6045, 26.473099914602905, 0.0, 0.0, 0.0, 4.269854824935952, 17.079419299743808,
0.8539709649871904, 0.8539709649871904, 1.7079419299743808, 5.977796754910333, 1.7079419299743808, 3.4158838599487615, 0.0, 3.4158838599487615, 0.0], [28.224455611390287,
5.412462900891187, 19.327272727272728, 10.117509625731555, 1063.0, 5754, 36.850921273031825, 0.0, 0.0, 0.0, 0.0, 12.562814070351758, 1.6750418760469012, 0.0, 0.0, 16.750418760469012, 0.0,
0.8375209380234506, 0.0, 1.6750418760469012, 0.0], [26.574633304572915, 5.159090909090909, 16.688524590163933, 10.637917002450935, 1017.0, 5346, 32.786885245901644, 0.0, 0.0, 0.0, 0.0,
10.353753235547886, 1.7256255392579811, 0.0, 1.7256255392579811, 8.628127696289905, 0.0, 3.4512510785159622, 1.7256255392579811, 2.5884383088869716, 0.0], [28.230088495575224,
5.592476489028213, 16.59016393442623, 8.013057068290658, 1012.0, 5736, 30.97345132743363, 0.0, 0.0, 0.0, 0.0, 12.389380530973451, 1.7699115044247788, 0.0, 1.7699115044247788,
15.044247787610619, 1.7699115044247788, 3.5398230088495577, 4.424778761061947, 3.5398230088495577, 0.0], [28.644067796610166, 5.455621301775148, 16.046153846153846, 8.769581639404329,
1043.0, 5879, 21.1864406779661, 0.0, 0.0, 0.0, 4.237288135593221, 11.016949152542374, 0.8474576271186441, 0.0, 6.779661016949152, 8.474576271186441, 2.542372881355932,
1.6949152542372881, 1.6949152542372881, 4.237288135593221, 0.0], [29.202779296360485, 5.649851632047477, 18.553571428571427, 8.115064569008906, 1039.0, 5972, 32.92894280762565, 0.0, 0.0,
0.0, 0.0, 14.731369150779896, 3.466204506065858, 0.0, 0.8665511265164645, 9.532062391681109, 1.7331022530329929, 2.5996533795493932, 0.8665511265164645, 5.1993067590987865, 0.0]],
["same", "same", "same", "same", "same", "same"]]], ["EN19", [[[31.180017226528854, 5.5220994475138125, 17.120689655172413, 8.646461477165088, 993.0, 5612, 56.84754521963824,

Figure A.2 - Sample data - Feature set I

A.2 Example: Feature extraction employing feature set II

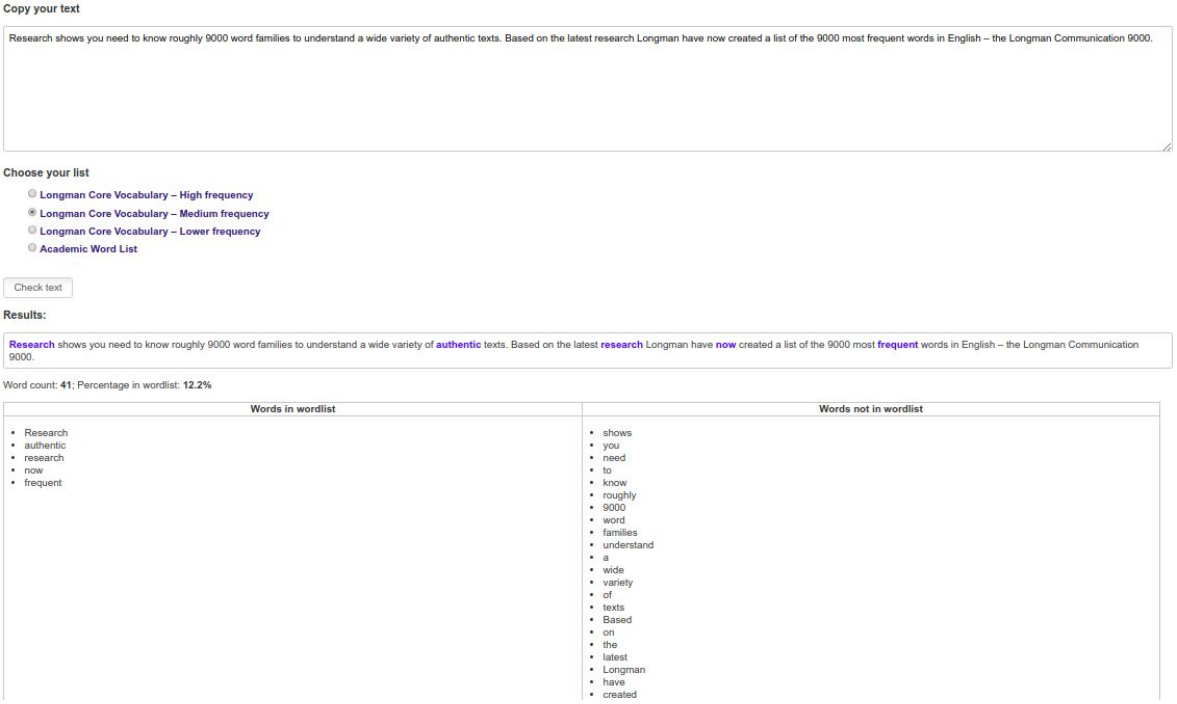

Figure A.3 - Feature extraction from documents - Feature set II

```
{"out": [
["EN04", [[[ 76.6, 18.9, 9.2, 15.8],[72.4, 15.9, 9.8, 16.5],[75, 20.1, 9.5, 17.5],[76.9,16.9, 10.1, 19.4], [77.1, 21.7, 7.6, 17]], ["same","same","same","same","same","same"]]],
["EN21", [[[75.6, 20.4, 6.7, 15.8],[73.4, 15.7, 7.9, 18.4], [72.3, 16.3, 4.4, 10.7]], ["same", "same", "not"]]],
["EN23", [[[75.6, 20.4, 6.7, 15.8],[73.4, 15.7, 7.9, 18.4], [73.5, 26.6, 6.2, 18.9]], ["same", "same", "same"]]],
["EN13", [[[69.9, 15.8, 6.7, 9.4], [76.1, 12.6, 8.1, 11.4], [73.6, 12.6, 8.1, 10], [60.2, 16.3, 5.5, 10.5]], ["same","same","same","not"]]],
["EN07", [[[76.6, 18.9, 9.2, 15.8],[72.4, 15.9, 9.8, 16.5], [75, 20.1, 9.5, 17.5], [76.9, 16.9, 10.1, 19.4], [75.7, 18.5, 7.5, 16.4]], ["same","same","same","same","not"]]],
["EN11", [[[77.9, 22.4, 5.7, 9.9], [68.5, 17.4, 6.3, 13.3], [60,20,6.5,16.1]], ["same","same","same"]]],
["EN19", [[[69.9, 15.8, 6.7,9.4], [76.1, 12.6, 8.1, 11.4], [73.6, 12.6, 8.1, 10], [70.9, 17.5, 6.1, 11.5]], ["same","same","same","same"]]],
["EN30", [[[77.9, 22.4, 5.7, 9.9], [68.5, 17.4, 6.3, 11.6], [68.7, 18.6, 5.6, 10.6]], ["same", "same", "not"]]],
["EE02", [[[90.4, 12.6, 3.6, 2.8], [83.1, 12.7,2.8,3.8]], ["same","not"]]],
["EE06", [[[86.8, 14.9, 3.2, 7], [77.9,11.6,3,5.4], [86.2, 17.4, 3.3,2.9]], ["same","same","not"]]],
["EE462", [[[ 87.8, 10.1, 2.3, 3], [85.3, 10.1, 2.3, 3]],["same", "same"]]],
["EE500", [[[88.7, 8.8, 2.9, 1.8], [82.6, 13.5, 4.3, 1.9], [ 90.5, 9.9, 2.5, 2.8], [86.2, 9.8, 4.4, 1.4], [84.1, 10.6, 1.5, 3.1]], ["same","same","same","same","same"]]],
["EE440", [[[81.6,16.8,5,3.9],[83.9,12.4,3.9,2.4],[79.1,16.6,5,5],[82.8,14.9,5.4,5.3],[82.4,13.2,4.8,5.2]],["same","same","same","same","same"]]],
["EE441", [[[83.8,12.2,3.9,1.6],[82.8,15.6,3.6,3.4],[81.4,14,6.1,4.8],[82.7,10.9,3.3,2.8],[81.1,12.2,5.3,3.7]],["same","same","same","same","not"]]],
["EE452", [[[81.8, 10, 2.7,2.7],[89.1,10.1,4.5,3.9]],["same","same"]]],
["EE454", [[[84.6,15.6,3.2,4.7],[83.2,13,5.3,5.2]],["same","same"]]],
["EE456", [[[91.3,13.6,2.1,1.3],[85.3,11.3,5.6,2.9]],["same","not"]]],
["EE459", [[[85.1,12.1,3.3,1.7],[92.2,12.9,3.2,2.1],[90.1,8.8,3.1,3.3],[81.3,10,2.9,1.9]],["same","same","same","not"]]]
]}
```


Figure A.4 - Sample data - Feature set II

Appendix B: Code Listings

B.1 Feature Extraction - Feature set I

```
# import libraries stylometry, nltk
from stylometry.extract import *
import nltk
from collections import Counter

# Function to extract 21 feature set from known author data
# Takes document path as argument and returns feature vector for the document
def extract_features(path):
    processedOutput = StyloDocument(path)
    processedOutput.text_output()

    return [processedOutput.type_token_ratio(),
              processedOutput.mean_word_len(),
              processedOutput.mean_sentence_len(),
              processedOutput.std_sentence_len(),
              processedOutput.mean_paragraph_len(),
              processedOutput.document_len(),
              processedOutput.term_per_thousand(','),
              processedOutput.term_per_thousand(';'),
              processedOutput.term_per_thousand('"'),
              processedOutput.term_per_thousand('!'),
              processedOutput.term_per_thousand('-'),
              processedOutput.term_per_thousand('and'),
              processedOutput.term_per_thousand('but'),
              processedOutput.term_per_thousand('however'),
              processedOutput.term_per_thousand('if'),
              processedOutput.term_per_thousand('that'),
              processedOutput.term_per_thousand('more'),
              processedOutput.term_per_thousand('must'),
              processedOutput.term_per_thousand('might'),
              processedOutput.term_per_thousand('this'),
              processedOutput.term_per_thousand('very'),
              ]
```

B.2 - Reading files and extracting features - Feature set I

```
# Import libraries to read files
from os import listdir
from os.path import isfile, join
import json
import codecs
import io

# Give folder path for dataset
folder_path = "/home/Desktop/Authorship Verfication/PAN/pan14-authorship-verification-training-corpus-2014-04-22/"

# Give path for json file to write or read the extracted features
with open('input.json') as data_file:
    data = json.load(data_file)

# Read data from input file if available
output = { "out" : [] }
output = data

# Open folders and read files
# Read folder names and actual labels from truth.txt file
with codecs.open(folder_path+"truth.txt", "r", encoding="utf-8-sig") as file:

    for i in file:
        a = i.split()
        folder = a[0]
        truth = a[1].replace('\n', '')   # Read actual labels for instance
        truth = truth.replace('\r', '')

        onlyfiles = [f for f in listdir(folder_path+folder) if isfile(join(folder_path+str(folder), f))]
        X = [] # Initialize feature vector for each document set
        y = [] # Initialize labels for each document set

        t = [folder,[]]
        output["out"].append(t)

        index = 1
        for k in onlyfiles:
            # Read each file in folder
            with io.open(folder_path + folder + "/" + k, "rU", encoding='utf-8') as f:
                # text = f.read().replace('\n', '')
                text = f.read()
                text = text.lower()

            path = folder_path + folder + "/" + k
            X.append(extract_features(text, path)) # Call function to extract features

            # Append necessary labels according to actual label
            if (truth == 'Y'):
                y.append('same')
            else :
                if (index == len(onlyfiles)):
                    y.append('not')
                else:
                    y.append('same')

            index = index + 1

        # Append feature vectors and labels to main vector
        t[1].append(X)
        t[1].append(y)

# Write extracted feature vector to file
with open('input.json', 'w') as outfile:
    json.dump(output, outfile)

file.close()
```

B.3 Training One-class SVM

```
# Import libraries
import json
from sklearn import svm
# Read feature vector from file
with open('input.json') as data_file:
    data = json.load(data_file)

data = data["out"]

overall_error = []
overall_correct = []

# For each author instance create SVM classifier
for inp in data:
    clf = svm.OneClassSVM(nu=0.325, kernel="linear", gamma=0.1)

    X = inp[1][0]
    y = inp[1][1]

    last = X.pop() # Feature vector of unknown document


    clf.fit(X) # Train classifier with training data of known author

    y_pred_train = clf.predict(last) # Predict label of unknown document

    # Mark True positives, false positives etc.
    if (y_pred_train[0] == 1) and (y[len(y)-1] == 'same'):
        overall_correct.append(1)
    elif (y_pred_train[0] == -1) and (y[len(y)-1] == 'not'):
        overall_correct.append(2)
    elif (y_pred_train[0] == 1) and (y[len(y)-1] == 'not'):
        overall_correct.append(3)
    else:
        overall_correct.append(0)

# Calculate evaluation matrices
total = len(overall_correct)
accuracy = (overall_correct.count(1)+overall_correct.count(2))/float(total)
precision = overall_correct.count(1)/float(overall_correct.count(1)+overall_correct.count(3))
recall = overall_correct.count(1)/float(overall_correct.count(1)+overall_correct.count(0))
f1 = 2/float((1/float(precision))+(1/float(recall)))
```

B.4 - Training Two-class SVM

```
# Import libraries
import json
from sklearn import svm

# Read feature vectors for two-class classification
with open('train.json') as data_file:
    data = json.load(data_file)

data = data["out"]

# Initialize variables to calculate evaluation matrices
overall_error = []
overall_correct = []
correct_count = 0
tp = 0
tn = 0
fp = 0
fn = 0

# For each instance of author create svm
for inp in data:
    clf = svm.SVC(kernel='linear', probability=True)

    X = inp[1][0]   # Feature vector
    y = inp[1][1]   # Input labels

    truth = inp[1][2] # Actual label of unknown document

    test = X.pop()  # Feature vector of unknown document

    # Initialize list to calculate mean of each feature vector of known documents
    val = [0.0]*21

    # Calculate mean value of feature vectors of known documents
    for i in range(0, len(X)):
        for j in range(0, 21):
            val[j] = val[j]+X[i][j]

    for i in range(0, 21):
        val[i] = val[i]/len(X)
```

```
    # Function to change mean feature vector and produce outlier feature vectors
    outliers, labels = deviate(val)

    # Append outlier feature vectors and labels
    X.append(outliers)
    y.append(labels)

    # Train model with combined feature vector for two classes "same" and "not"
    clf.fit(X, y)

    # Predict label for unknown document
    prediction = clf.predict(test)

    # Calculate true positives, false positives etc.
    if prediction[0] == "same" and truth == 'Y':
        tp = tp + 1
    elif prediction[0] == "not" and truth == 'N':
        tn = tn + 1
    elif prediction[0] == "same" and truth == 'N':
        fp = fp + 1
    elif prediction[0] == "not" and truth == 'Y':
        fn = fn + 1

# Calculate evaluation matrices
correct_count = tp + tn
precision = fp/float(tp+fp)
recall = fp/float(tp+fn)
accuracy =float(tp+tn)/(tp+tn+fp+fn)
f1 = 2/float(((1/float(precision))+(1/float(recall))))
```