ENSEMBLES OF MACHINE LEARNING MODELS FOR DETECTING WRITING STYLE CHANGES AT THE SENTENCE LEVEL

BY

OLOO VIVIAN ANYANGO

PHD/CI/00064/017

A THESIS SUBMITTED IN FULFILMENT FOR THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING AND INFORMATICS

MASENO UNIVERSITY

DECLARATION

I declare that this thesis is my original work and has not been presented for an award of a degree in any university.

Vivian Anyango Oloo

PHD/CI/00064/017

Signature……………………… Date……………………

This thesis has been submitted for examination with our approval as university supervisors

Dr. Lilian Awuor Wanzare

School of Computing and Informatics

Maseno University

Signature……………………… Date……………………

Dr. Calvins Otieno

School of Computing and Informatics

Maseno University

Signature……………………… Date……………………

ACKNOWLEDMENT

The completion of this work was a concerted effort of various groups of people who's contributions are sincerely appreciated. My special thanks go to my supervisors; Dr. Lilian Wanzare and Dr. Calvins Otieno both of Maseno University for the mentor-ship and technical support throughout the study. Secondly, I am grateful to the entire School of Computing faculty for tirelessly providing technical inputs, wealth of knowledge and sufficient guidance this far.

I am equally grateful to Zenodo and the PAN technical committee for providing the data set that was used in the study. I am indebted to my children Kirstein, Bill and Griffin, and my husband Isaac for the love and support. I cannot forget my late dad Oloo, mum and all my siblings for their continuous encouragement, patience and love.

I also thank the Almighty God for granting me health and grace to complete this study.

DEDICATION

This work is dedicated to Kirsten, Bill, Griffin, my husband Isaac and late father Joseph Oloo.

ABSTRACT

Establishing the exact number of authors collaborating in writing a document is the focus of writing styles change detection models. However, existing writing style change detection models fail to adequately detect writing style changes in documents where each author writes very short texts in form of sentences, which are randomly distributed in the document. In addition, a number of features have been used in detecting writing styles but few studies have determined their suitability for this task. For writing style change detection models to remain relevant, there is need for models that can detect writing styles changes at the sentence level. The aim of this study was to develop ensembles of machine learning models for detecting writing style changes at the sentence level. The specific objectives were; to design ensembles of machine learning models for detecting writing style changes in documents, to implement ensembles of machine learning models for detecting writing style changes, to determine optimal feature sets for detecting writing style changes, and to evaluate the effectiveness of the ensemble models on detecting writing style changes at the sentence level. The study variables were the ensembles of machine learning models, while the dependent variable was the detection of writing style changes at the sentence level. Other variables looked at were the feature sets, model evaluation at the sentence level and performance of the model on detecting writing style changes at the sentence level. Mixed research design was used in this study, where exploratory design was used to identify stylometric features for use in the study. Features whose importance scores were greater than zero were considered optimal and were used to carry out experiments. Under experimental design, four experiments were performed: first to select the optimal document features and second to select the optimal sentence level features using feature importance scores. The third experiment was designed to classify documents as either single authored or multi-authored. The last experiment was used to detect the number of writing style changes in documents classified as multi-authored. The Pan at Clef 2019 style change date set was used to train, validate and test the models. The corpus consisted of 5088 documents out of which 50% was used for training, 25% for validation and 25% for testing. Half of the documents were single authored while the other half were multi-authored. Results show that 19 features were optimal at the document level while twenty two features were optimal sentence level. The models were able to classify single authored documents and multi-authored documents with an accuracy of 0.91 and an F1score of 0.90. Similarly, the study achieved an Ordinal Classification Index of 0.731 in detecting the number of writing style changes in multi-authored documents outperforming state-of-the-art models which achieved 0.808. The better performance is attributed to the use of optimal feature sets, ensembles learning models and sentence level representation. The main contribution of this study is ensembles of machine learning models able to detect writing style changes at the sentence level. In addition, the study identified two sets of features; the optimal document and sentence level feature sets which can be used for writing style change detection with improved performance.

TABLE OF CONTENT

LIST OF ABBREVIATIONS AND ACRONYMS

AA	:	Authorship Attribution
SVM	:	Support Vector Machines
LR	:	Logistic Regression
OCI	:	Ordinal Classification Index
KNN	:	K- Nearest Neighbors
MUERC	:	Maseno University Ethics Review Committee
BC	:	Before Christ
POS	:	Part of Speech
NLTK	:	Natural Language Toolkit
HTML	:	Hyper Text Markup Language
URL	:	Unified Resource Location
TF-IDF	:	Term Frequency Inverse Document Frequency
BERT	:	Bi-directional Encoder Representation from Transformers
ELMs	:	Extreme Learning Machines
RF	:	Random Forest
BIRCH	:	Balanced Iterative Reducing and Clustering Using Hierarchies
PAN	:	A series of Scientific Events and Shared Tasks on digital text forensics and stylometry
EER	:	Equal Error Rate
RNN	:	Recurrent Neural Networks
CLEF	:	Conference and Labs of Evaluation Forum
CERN	:	European Organization for Nuclear Research

OPERATIONAL TERMS

Accuracy: Refers to the percentage of the correct predictions.

Authentication: Verification of claimed identity

Authorship Attribution: This constitutes the identification of the most likely author of a document from a list of known writers.

Author Verification: This refers to the process of identifying whether anonymously written documents have the same authorship styles.

Authorship Profiling: Refers to predicting the demographic attributes of authors based on the styles of writing.

Consensus Function: A method of selecting the final results of ensemble learning from the outputs of individual algorithms of the ensemble.

Ensembles of Supervised Model: A model of three supervised machine learning algorithms combined together using majority voting.

Ensembles of Clustering Model: A model consisting of three unsupervised machine learning algorithms combined together using the median partition-based consensus function.

K-means Clustering: A clustering algorithm based on the Euclidean distance measure used to cluster objects based on how far the object is from the cluster centroid.

Multi- Author Analysis: This is how the number of authors who worked together to write a text is determined.

Ordinal Classification Index: A type of multi-class categorization in which the classes are arranged in a certain way but do not significantly differ numerically.

Precision: Refers to the ratio of correct predictions relative to all the documents in a group.

Recall: This is used to refer to the ratio of documents that are correctly clustered as written by an author to the total number of documents written by that author.

Short text: Piece of text having not more than 500 words.

Stylometry: This refers to the quantization of writing style characteristics for purposes of authorship identification.

Writing style: The way an author expresses his/her self in written text characterized by the unique choices of words and other attributes that are traceable in all his/her works.

Writing style change detection: Determining authorship changes in a document by analyzing the writing styles of individual authors and evaluating the differences in writing styles applied in a document.

Sentence Level: Refers to the structure and formation of a sentence.

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

Humans have been involved in writing for centuries now in order to pass their ideas and thoughts to others. Writing was invented between 3400 – 3200 BC, independently in four countries: Mesopotamia (the present-day Iraq), Egypt, China and Mesoamerica (Millard, 2006). At the initial stages writing was used to count and several objects with different shapes were developed to aid this process. This counting continued for about 1000 years up to the 2001 BC when there was a paradigm shift to the use of script. Since then, writing has evolved from these traditional methods to the present day use of the alphabet (Dobao, 2015).

Writing is the way an individual expresses his/herself using symbols. It is the process of communicating information such as ideas and thoughts to others in organized ways and in a form recognizable by them. According to Besserat and Erard, (2009) writing involves the entire process of information collection, processing and dissemination. It involves choosing which symbols to use from the numerous symbols and their combinations to form words, sentences and paragraphs, able to relay the intended information. According to Romanov et al., (2020) at any given time each writer has a subconscious habit of using words depending on the subject matter, the genre, the supposed audience, level of education and age. These choices are unique for every individual, and can represent an author’s style of writing.

A writing style constitutes persistent choices of words and other symbols that are unique to each and every individual and that are traceable in all their works (Anwar et al., 2019; Ramnial et al., 2016; Brocardo et al., 2013). Writing styles can be learned and quantified using stylometric features, and can be used to establish an author’s stylistic signature applicable in discriminating between works of different authors (Juola, 2006). Whereas the use of stylistic signature is still debatable, recent studies in stylometry using machine learning indicate the possibility of success in defining stylistic signatures (Nath, 2019, Castro-Castro et al. 2020).

Stylometry defined as the study of writing styles, examines the writing styles of authors for specific attributes able to discriminate between two or more works. It uses stylometric features to distinguish between writing styles of authors. It has been used in authorship analysis studies for authorship attribution, authorship verification and authorship profiling. There exist several features under stylometry that can be used to uniquely identify authors. These features are applicable as either standalone e.g a single feature or as feature combinations of two or more features in authorship studies (Anwar et al., 2019; Akiva & Koppel, 2012). under stylometry, the study discussed stylometric features and authorship analysis in the following sub-sections.

1.1.1 Stylometric Features

There are several stylometric features, over 1000, defined in literature. These features can be categorized into three, four and five categories. Categorization of features into three categories groups all stylometric features into either syntantic, lexical and content-specific features. When four categories are defined, stylometric features fall either into lexical, syntactic, structural and content features. While five categories breaks down the lexical feature category into lexical and character features, syntactic, structural, content-specific and character features. Other studies have also proposed certain idiosyncratic features for authorship analysis.

Different features have been applied in author verification studies, with each feature category yielding variable results depending on the environment and the composition of the data set. For instance, lexical features have yielded better results in studies involving different languages while syntactic features yield better results when the stylistic difference in different works is the main target (Gómez-Adorno et al., 2018; Akiva & Koppel, 2012; Houvardas & Stamatatos, 2006b, 2006a; Naga Prasad et al., 2015; Zheng et al., 2006). In situations where the contextual biases are the key focus, some studies have applied content-specific features and structural features with success.

Lexical features can represent partiality of authors of using certain words or string of character in all their works. They are considerably the most popular feature category in previous studies because they are applicable to any language with no extra cost and requirements. This characteristic makes them good candidates for authorship verification studies since they can measure the stylistic differences between writers ignoring the differences in languages (Shrestha et al., 2017; Abbasi & Chen, 2008; Brocardo et al., 2015; Ding et al., 2016; Houvardas & Stamatatos, 2006b; Howedi et al., 2020; Howedi & Mohd, 2014). Some examples of these features include pronouns, adjectives, nouns, word n-grams etc (see section 2.4.1.1).

Syntactic features are the features that provide the overall sequential structure. Punctuation, function words, verb phrases, phrase types are some of the widely used syntactic features in authorship verification. These features are considered the most appropriate authorship stylistic signature because they are employed unconsciously by authors (Anwar et al., 2019; Castro-Castro et al., 2020; Sari, 2018; Koppel & Schler, 2004). Syntactic features include part of speech tags, punctuation and functional words. Functional words are words that are used to explain or create grammatical or structural relationships into which the content words may fit. They have no lexical or semantic content on their own. Examples are conjunctions, determiners, prepositions and pronouns (see section 2.4.1.3).

Structural features are used to define the general document organization by individual authors. They include features like length of sentences in a document, the length of words, size of fonts used, font color etc.(Abbasi & Chen, 2008; De Vel et al., 2001; Houvardas & Stamatatos, 2006a, 2006b; Naga Prasad et al., 2015; Zheng et al., 2006). Some commonly used structural features are font style, font color, use of tab spaces, paragraph alignments, spacing style, length of sentences and length of words are features which define the general organization of a document. The preference to certain font style or font coloration can be used to model an author’s style. Similarly, the choice of the various spacing styles or the general preference to a specific spacing style can indicate works of a specific author (see section 2.4.1.4).

Content-specific features a signal extant of particular key words, interest groups and given activities. They are particular and issue contextual information of the task at hand. For instance, Jiexun et al., (2006) manually observed, analyzed texts written in ancient times and identified some key words in the context of online sale environment such as obbo, windows, hashtags, etc (Abbasi & Chen, 2008; Kaur et al., 2020; Ramnial et al., 2016; Zheng et al., 2006).The most frequently used context features are frequency of emoticons, forwards and tagging, number of links to other pages and hash tags (see section 2.4.1.5).

Character features are used to capture disparities in the use of lexical information, such as capitalization and punctuation. Among these are character n-grams, the total amount of characters, the total number of numbers, the total number of capital letters, the total number of space characters, the number of tabs and their corresponding ratios, and the use of special characters. Tokenizing words into characters allows for the extraction of character features. Character features have been used in previous research either alone or in an ensemble (see section 2.3.1.1).

Application of stylometric features come in two-fold; uni-variate and multivariate analysis. In uni-variate studies, the authorship verification analysis is based on determining the effect of a single feature item, while multivariate techniques involve the use of ensemble of features and their effect on authorship verification tasks. Although uni-variate studies continue to yield good results especially in long documents, there is consensus that no single feature is adequate in distinguishing between the works of several authors (Argamon et al., 2007; Houvardas & Stamatatos, 2006; Juola, 2006).

Recent studies recommend the use of ensemble methods when using short text length in authorship verification (Castro-Castro et al., 2020; Das et al., 2018; Shrestha et al., 2017; Zuo, Zhao & Banerjee, 2019). However, studies have not agreed on an optimized feature set for authorship verification applicable to all persons and domains. Therefore identifying the most appropriate feature set combination for specific tasks, still considered challenging, is an important task in authorship analysis studies. This study considered variety of features in an ensemble method of features for the writing style change detection task since they have been shown to yield better performance as opposed to the uni-variate methods.

1.1.2 Authorship Analysis

Authorship analysis encompasses three main tasks; author identification, verification and profiling of authors. Authorship identification, involves identifying who the author of an anonymously written document is from a group of known authorship (Anwar et al., 2019; Juola, 2008: Sari, 2018). It has been used to solve authorship disputes among authors or to ascertain the most probable author of unknown text. For instance, Mendehall (1887) and Mascol (1888) analyzed the distribution of sentence and word lengths, to determine the disputed plays between Shakespear and Bacon. While Monsteller and Wallace, (1964) attempted to establish the most likely author of the twelve disputed paper in a bid to solve the dispute between Alexander Hamilton, James Madison and John Jay.

Traditionally, authorship attribution was employed in closed-set scenarios where the authorship of the document under question was among the training sample (Anwar et al., 2019). In addition, these studies analyzed very long documents on fairly small sample sizes that have been shown to yield very high accuracy. However, state of the art models try to solve more practical problems which include the use of shorter texts in open-set scenarios where the authorship of the document in question is not known in advance (Rexha et al, 2018; Jasper, 2018; Sittar et al, 2018). Although authorship attribution is now considered a simple task, identifying authors of short text lengths is still considered a difficult task.

Author verification can be defined as the confirmation of whether two texts A and B of unknown authorship were written by the same author (Potha & Stamatatos, 2018). It has been applied in plagiarism detection and other algorithms for maintaining online academic integrity. Whereas extensive research has been conducted in authorship attribution, author verification has not received the same focus (Brocardo et al., 2015). Early studies on authorship verification focused on scenarios where there are several documents and just one author. Existing models adopt a real-world approach where there are several documents and more than one author.

As opposed to the simple task of identifying authors, authorship verification is a difficult undertaking since there could be slight variations on the authorship of individual writers with the passage of time (Can & Patton, 2004; Gómez-Adorno et al., 2018). Moreover most studies assume that the entire document is written by just one author, with few studies investigating the case of multi-author authentication (Koppel & Winter, 2014; Zhou & Wang, n.d.). These studies assume a scenario where the number of author-ships is known in advance. Authorship verification has also been investigated with long and short text. However, it is reported that authorship verification on shorter text is still challenging (Barlas & Stamatatos, 2020; Brocardo et al., 2013, 2015). Profiling involves analyzing characteristics of an author of an anonymous document for the purpose of creating authorship profiles (Ramnial et al., 2016). Authorship profiling has been used to predict the gender of the authors of particular documents using relief features.

1.1.3 Challenges in Authorship Analysis

Most of the existing authorship analysis strategies have been used in simple authorship attribution and verification tasks that assume that each document has only one author(Anwar et al., 2019; Can & Patton, 2004; Gómez-Adorno et al., 2018; Juola, 2008; Sari, 2018). In real world scenarios where several authors may participate in writing a single, authorship verification can still suffer from the following challenges; authorship verification using short text length, individual author writing style change over time and multiple authorship of a document. A document of lengths ranging below 500 characters is considered a short document in authorship analysis. Verification of shorter documents is quite challenging because of inadequate training data to differentiate between the writing styles of authors giving rise to similarity overlap problems evident in most authorship verification studies (Castro-Castro et al., 2020; Nath, 2019; Zuo, Zhao & Banerjee, 2019). Brocardo et al., (2013) used an ensemble of supervised learning and syntactic features to determine the authors of online text. They applied stylometric techniques on a corpus of 87 authors from the Enron emails. The approach of Brocardo et al., (2013) concluded that there is no single feature that can sufficiently discriminate the styles of writings of different authors. They opine that ensemble of features would yield better results.

Individual author style change over time refers to evolutionary changes in an author's writing style over time. Change in writing style over time is a natural phenomenon whose effect on the overall performance of authorship verification cannot be overemphasized. A few studies have considered change in writing style with time using various methodologies and data sets. For instance, Lancashire & Hirst, (2009) used text analysis computing tools vocabulary richness feature on Agatha Christie's novels to determine the possibility of variations in her style of writing, an indication of Dementia. Gómez-Adorno et al., (2018) studied writing styles of 7 authors of English novels downloaded from project Guternberg. They subdivided the novels into three groups corresponding to the authorship stage of an author: early, middle and late, with a difference of three years between each group. They found out that writing styles of authors can change significantly in a span of three years.

Multi-author analysis, referred to as determining the number of authors in a document, has been studied by detecting changes in writing styles in documents. The task of detecting change of writing styles involves the determination of whether a document is single-authored or multi-authored, and determining locations where authorship change. Writing style change detection is now the focus of many authorship analysis studies and the most challenging task (Kumar et al., 2019; Zangerle et al., 2019). For instance, Akiva & Koppel, (2012) used supervised (linear

SVM) learning to determine the authors of a multi-authored document. The feature set used consisted of 500 most common words in a document. This approach yielded accuracy of 88-96% for author pairs while documents with three authors obtained accuracy ranging 77-82%. Documents with four authors obtained a purity of 74%. They observed a decline in verification accuracy when the number of authors increased.

Writing style change detection has been the focus of PAN tasks lately. PAN is an organization that defines a series of scientific events and shared tasks on digital text forensics and stylometry. They organize competitions and conferences dubbed, Conference and Labs of the Evaluation Forum (CLEF) where participants present their works and papers. In addition, they provide data sets for the various tasks of authorship verification, authorship attribution and style change detection. Writing style change detection is part of authorship analysis that encompasses authorship attribution and verification, facilitated by the PAN CLEF author verification competitions. They have defined a number of author verification tasks as stated in PAN 2017, PAN 2018, 2019, 2020 and the current PAN. Detecting changes in writing styles in documents has the ultimate goal of identifying the total number of authors participating in writing a document, and locating specific positions where authorship changes within a collaborative writing. Despite the various studies conducted in this area under PAN competition, Zangerle et al., (2019) reports that it is still a challenging task.

Currently, authorship verification studies focus on checking authorship of short text, writing style change detection, and change of writing styles with time. Whereas numerous studies continue to yield promising results in these areas, no study has been able to satisfactorily address these challenges. For instance, in multi-author analysis it is reported that as the number of authors of a text grows, verification accuracy decreases (Akiva & Koppel, 2012). Text length on the other hand presents an opposite behavior; verification accuracy decreases with a decrease in text length and vice versa (Brocardo et al., 2015).

This study focused on determining the number of authors in a text, by identifying the number of style changes. First the optimal feature sets were determined for the two tasks; separating single authors from multi-authors and determining the number of writing style changes in documents classified as multi-authored. For the first task, the study used the optimal document level features to classify documents as either single or multi-authored. The second task involved using the optimal sentence level features to determine the number of writing style changes in multi-authored documents. Two models were designed and developed: an ensemble model of supervised learning classifiers and an ensemble model of unsupervised learning clustering algorithms. Supervised learning models which are superior to unsupervised learning in scenarios where there is labeled data was used to separate single authored documents from multi-authored documents. The focus here was to have as many documents correctly classified as possible and hence the need for supervised learning. On the other hand unsupervised learning was used on multi-authored documents because of inadequate labeled data, and because the study focused on real-world scenarios where labeled data is hard to come by. The ensemble model of clustering; K-means, Gaussian Mixture Models and Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) models was used, the results of which were passed through a consensus function to give the final results. Ensemble model provide better results by maximizing on the strength of individual algorithms. A number of features were examined including character features, lexical and syntactic, structural and content feature categories. It was believed that increasing the number of features improved the performance of the algorithm.

1.2 Statement of the Problem

Whereas multi-authorship continues to gain popularity, there are few writing style change detection models able to establish the number of authors in multi-authored documents. Existing models are not able to adequately detect the number of authors in documents where each author writes very short texts, in form of sentences, which are randomly distributed in the document.

However, in the real world, multi-authored documents may be written by many authors each contributing very short texts that are unevenly or randomly distributed in the document. Such short text contributions could be ignored by existing writing style change detection models which define writing style changes at the paragraph level. If writing style change detection models are to remain relevant, consistency of results is needed regardless of the text length and the number of authors.

1.3 Objectives

The main objective of the study was to develop ensembles of machine learning models for detecting writing style changes at the sentence level.

1.3.1 Specific Objectives

The following specific objectives were investigated to realize the main objective;

i. To design ensembles of machine learning models for detecting writing style changes.
ii. To implement the ensembles of machine learning models for detecting writing style changes
iii. To determine optimal feature sets usable by the ensembles of machine learning model for detecting writing style changes in documents.
iv. To evaluate the effectiveness of ensembles of machine learning models in detecting writing style changes at the sentence level.

1.3.2 Research Questions

The following research questions were answered by the study:

i. How will the ensembles of machine learning models be designed?
ii. How will the ensembles of machine learning models be implemented?
iii. How will the optimal feature sets for detecting writing style changes in documents be determined?
iv. How does the performance of the ensembles of machine learning models compare with existing models for detecting writing style changes at the sentence level?

1.4 Significance of the Study

The study contributed to the field of writing style change detection by conducting and publishing a survey on state-of-the-art writing style change detection approaches. This survey summarized strengths and weaknesses of existing writing style change detection models, information which is useful in furthering research in this area. The outputs of this study were ensembles of machine learning models able to detect writing styles changes on short texts, sentences, using a variety of features. Such models can find practical applications in forensic investigations where the suspects may write very short texts to change the contents of a document or the flow of operations. This study determined and published optimal document and sentence level feature sets for the writing style change detection. These features can be used to determine whether a document is single authored or multi-authored, and determining the number of writing style change detection in documents.

Writing style change detection on short texts larger number of authors is an important exercise that could have its application in academic integrity preservation especially in institutions of higher learning, to mitigate academic indiscipline or corruption. A part from affecting the reputation of institutions of higher learning and the academic awards they give to students; academic corruption may have very serious consequences on a nation or the entire world if left unattended.

1.5 Scope of the Study

This study focused on the English language only and therefore other languages were considered out of scope. In addition, only writing style change detection on short text in cross-genre scenarios was considered. Evolutionary changes in authors' writing styles over time and change in writing styles as a result of imitation, standardization and obfuscation of writer's style were considered out of scope of this study.

The study used ensembles of machine learning models because of their performance strengths compared to the use of individual algorithms. Bagging ensembles which tend to reduce variance hence eliminating over-fitting in models was used. The other types of ensemble learning were considered out of scope.

1.6 Limitations of the Study

The main challenges this study faced were two-fold. First, the length of the training data was a challenge because the entire document was split in its constituent sentences, which was then used to represent an authors' writing style. The writing style change model - the clustering model, was trained on features extracted at the sentence level thereby reducing the significance of these features on discriminating among different writing styles. Machine learning models require a lot of data such as a paragraph to be able to learn the style of writing of an author and to make selected features significant in separating between two or more style. However in this study, the document length on which the models were trained was a sentence and thiscould affect the performance the models.

The second challenge faced was the high computational requirements. Specifically, the models required a lot of memory space and higher computational power processors. For instance, K-means required that the entire data set first reside in memory before the model is executed. GMMs and BIRCH use subsets of the data set to create models which then are used to perform predictions. The overall model prediction is obtained as a result of doing several runs on the model either with the entire data set as in the case of K-means or with different subsets of data as with GMMs and BIRCH. To ensure consistency of results, several runs are required, over ten runs. However, due to the limited memory the number of runs was limited to the first instances of consistent results. Moreover, the software used, the Python 3.9 version could not run on windows 7 or earlier version, necessitating that more current versions of windows be used. Originally, the software tools which were available were windows 7, but because it could not support Python 3.9 the study had to upgrade it to windows 11. This venture added more cost to the study.

The study limited the number of base algorithms to three in each case because using a bigger number would result into more computing resource requirements, having more than 3 algorithms would require more computing resources and more training time thereby affecting the study which was constrained by time and resources.

1.7 Assumptions

It was assumed that an entire sentence was the work of one author and therefore the models were trained on this.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This chapter reviews literature on existing stylometric features and feature selection strategies, the state-of-the-art authorship verification models and the methods employed in writing style change detection.

The rest of this chapter is organized as follows: Section 2.2 discusses the existing models for writing style change detection, Section 2.3Outlines the Review of performance of base learners, Section 2.4outlines Determination of optimal feature sets for writing style change detection, Section 2.5 discusses evaluation metrics and performance of related models on writing style change detection, Section 2.6 the study outlines the gap, and Section 2.7 presents the conceptual framework while Section 2.8 summarizes existing work.

2.2 Existing Models for Multi-Author Analysis

Multi-author analysis began with the simple task of grouping together documents written by one author known as author clustering (Rosso et al., 2016), and author diarization which groups together sections of a document with the same writing style (Kocher, 2016; Kuznetsov, 2016; Safin & Kuznetsov, 2017). Author clustering assumes that an entire document has a single author and exploits the stylistic similarities and differences among documents to group documents with the same writing styles together. It can be seen as an adaptation of the conventional authorship verification which examines if two documents exhibit similar writing styles (Abbasi & Chen, 2005; Halvani & Graner, 2017). Author diarization on the other hand breaks a document into homogeneous sections representing similar writing styles. These basic multi-author analysis tasks form the basis of the writing style change detection.

Pioneer studies in the multi-author analysis assume that a document has one main author who writes about 70% of the document and the other authors contributing the rest of the sections. In this scenario the first few paragraphs are assumed to have been written by the main author and the rest of the other authors writes the remaining paragraphs (Castro-Castro et al., 2020). Here, the number of authors is determined by breaking down a document into sentence groups or paragraphs. The first paragraph is assigned to have been written by the main author and is compared with the preceding paragraphs to determine whether they are similar. If they are similar, then they have the same author and vice versa. These initial endeavors have expanded to include the task of separating single authored from multi-authored, determining the borders where authorship changes in multi-authored documents (Safin & Kuznetsov, 2017; Strom, 2021).

Further explorations on multi-author analysis include determining the number of authors in collaborative documents (Zangerle et al., 2019), and identifying whether there is style change between consecutive paragraphs, known as writing style change detection. Other tasks of writing style change detection include finding all positions of writing style change detection within a multi-authored document and assigning all paragraphs of the text uniquely to some author out of the assumed number of authors in the document (Nath, 2021; Zangerle et al., 2019, 2020).

Different scenarios can be defined with multi-authored documents; Firstly, the case of one main author and several small authors. Since there is one main author contributing a huge portion of the texts in the documents, studies have employed the use of outlier and anomaly detection methods, and hashing-based clustering to determine the number of authors in the document as in style breach detection and intrinsic plagiarism detection (Karas et al., 2017; Kuznetsov et al., 2016). Here the task is to find sections of the multi-authored documents which are not written by the main author and to label them as either ‘plagiarized’ or ‘outlier’.

The second scenario is the case of several small authors contributing texts randomly in the document. The task in this case is to determine the total number of authors by determining the similarities in the texts such as in paragraphs, sentences or sentence groups (Safin & Kuznetsov, 2017; Strom, 2021; Safin & Ogaltsov, 2018). This task can be challenging if many small authors are contributing relatively short texts due to similarity overlap (Brocardo et al., 2013, 2015; Castro-Castro et al., 2020). Attempts to solve this challenge include the use of clustering algorithms that groups together texts written by the same author. It is believed that a cluster contains only text written by the same author. Determining the optimal value of k (clusters) is the main challenge for these methods (Zuo, Zhao & Banerjee, 2019).

Writing style change detection is based on generation of feature vectors to be used to discriminate or group together documents. Feature vectors can be generated at the document level, paragraph, sentence and word levels. Document level feature generation is used in a case where different documents are to be grouped together or compared for similarities (Kuznetsov et al., 2016; Safin & Kuznetsova, 2017). Since there is sufficient data, reduced feature sets tend to yield better results in terms of runtime (Alberts, 2017; Kocher, 2016; Potha & Stamatatos, 2018). Document level feature generation has mostly been applied for the tasks of author clustering and separating single authored documents from multi-authored documents. Sentence level feature generation has also been used in writing style change detection particularly in multi-authored documents.

Feature vectors generated at the sentence level may result in higher purity because they may capture all the stylistic changes within a document including very short text contributions by other authors which may however be ignored (Kuznetsov et al., 2016; Safin & Kuznetsova, 2017). The main challenge with this method is that the feature set should be expanded so as to adequately represent an author’s writing style. Other studies combine a number of sentences together to form sentence groups, and generate feature vectors based on these groups. This may be seen as the most probable approach as it may provide a sizable amount of data for the style change detection task. However, it's limited since it may ignore very short text contributions made by other authors, such as a sentence contributed by another author or even a word, leading to reduced reliability of writing style change detection methods (Brocardo et al., 2015; Juola, 2006).

Several methods have been proposed to solve the problem of writing style change detection, this study discussed the different methods under author diarization and clustering and style change detection as indicated in the following subsections.

2.2.1 Author Diarization and Clustering

Author clustering aims to identify and group documents written by the same authors together while author diarization identifies parts of a multi-authored document written by the same (Rosso et al., 2016). Simple supervised learning methods such as decision trees have been used to generate feature vectors where labeled data is available (Kuznetsov et al., 2016), while unsupervised learning methods such as k-means are applicable in cases where only unlabeled data is available (Sittar et al., 2016). To solve the clustering task, feature vectors are generated at the document level so that similarities between document pairs are determined for placement in various clusters. Author diarization on the other hand generates features at either the sentence level or sentence group level (Kocher, 2016; Kuznetsov et al., 2016; Safin & Kuznetsov, 2017).

For author diarization, feature generation at the sentence level can be considered ideal since it may take care of even very short text contributions by other authors thereby improving the purity of these methods (Ramnial et al., 2016). However, this method may require the use of various combinations of features to be able to distinguish between works of different authors. Paragraph level feature vector generation seems to be practical as it can be assumed that a new author in a multi-authored document may have to contribute a number of sentences summing up to a paragraph for him/her to put across his/her train of thought (Kocher, 2016).

Several stylometric features types have been used in author diarization and clustering. Most studies employ the use of feature combinations such as lexical, syntactic and character features to analyze the variance in the styles of writing by different authors (Brocardo et al., 2013, 2015; Safin & Kuznetsova, 2017). In literature stylometric features such as vocabulary richness, word frequencies, sentence length in characters, mean sentence length, average word length, total number of words, ratio of interrogative sentences, character count, digits count, uppercase letters count, spaces count, tabs count, ratio of uppercase letters, ratios of spaces, ratios of tabs, frequent punctuation and Part of speech tags, function words, stop words, spelling mistakes, have been in author diarization and clustering (Kocher, 2016; Kuznetsov et al., 2016).

Feature combinations have been shown to produce better results when the text length is short as in author diarization (Brocardo et al., 2015; Juola, 2006). In addition, it has been shown that these features produce the best results in most authorship analysis studies. For instance, lexical features can be tokenized and can be quantified to an author’s writing style, while character features can be applied where text length is short. Syntactic features on the other hand are seen as the best feature type as the same attributes are applied subconsciously by a user throughout their writing (Abbasi & Chen, 2005; Safin & Ogaltsov, 2018).

Once the feature vectors have been generated, distance measures are then used to place documents or text in clusters. The idea is to form different clusters representative of the number of authors, and place each document/segment into exactly one cluster (Rosso et al., 2016). The distance measures are used to calculate the inter-cluster and/or intra-cluster distances for similarities and differences based on a predetermined threshold (Alberts, 2017; Garcia et al., 2017; Adorno et al., 2019). Documents are placed in a cluster if the distance between it and other documents in the cluster does not exceed a predefined value. Several distance measures have been proposed for the authorship clustering and diarization studies.

For instance, Kocher, (2016) used a simple distance measure called SPATIUM-L1 based on the L1-norm to cluster documents and pieces of text together. SPATIUM-L1 calculates the distance between a pair of sentences and places them in the same cluster if the threshold value is not exceeded. Sittar et al., (2016) used a cluster distance approach which they referred to as CLUSTDIST. The CLUSTDIST approach calculates the average distance of one portion of text to all other pieces of text, and places the portion in a different cluster if its distance from the other portions is greater than the average distance of all the texts in that cluster. Although the distance measures used in literature are simple, they yield comparable results to state-of-the art methods.

Author diarization and clustering has been solved by using outlier and anomaly detection techniques proposed by Kuznetsov et al., (2016) and Sittar et al., (2016). These methods rely on the assumption that one author writes the better part of the document, up to 70%, and the rest of the document is written by several authors who contribute short texts. In addition, the first few paragraphs in the document are contributed by the main author (Rosso et al., 2016). These methods generate a feature vector containing the average distances of all groups of texts from each other. The distance between a pair of feature vectors generated at the document or sentence level is calculated to see its deviation from other sentences or documents. Sittar et al., (2016) used a ClustDist anomaly detection technique on 15 lexical features to generate a feature vector containing average distances of all sentences from each other. The ClustDist method computes the distances between any pair of vectors. The resultant score for each sentence distance from others, generates a ranking which describes the deviation of a sentence from other sentences in the given document.

On the other hand, threshold-based outlier detection methods which are based on detecting outliers in an authors' style statistics have been investigated by some studies for their effectiveness in authorship clustering (Kuznetsov et al., 2016). Here the focus is identifying segments in the document which are not written by the main author (Nath, 2021; Ramnial et al., 2016). Kuznetsov et al., (2016) proposed an intrinsic plagiarism detection approach based on gradient boosting regression trees with optimal parameters set at n-estimators= 200 and max-depth = 4. This algorithm is based on threshold-based outlier detection for detecting outliers in an author’s style statistics to provide the label “plagiarized” to the outliers.

2.2.2 Writing Style Change Detection

Writing style change detection is the act of examining a document to identify the different styles of writing present in it (Rosso et al., 2016). The ultimate goal of style change detection is to determine the number of authors in a document and the various parts of the documents each author has contributed (Deibel & Lofflad, 2021; Vetter, Sakti & Nakamura, 2019). Research in this area is still slow because of the limited benchmark data sets and the limitations of machine learning algorithms on short length text (Brocardo et al., 2013, 2015; Castro-Castro et al., 2020). However, annual PAN competitions have contributed immensely to the growth of research in this area by providing benchmark data sets and defining tasks to be solved for the style change detection problem. Pioneer studies in style change detection focused on determining the number of authors in a document where it is believed that an author writes a considerably big chunk of text, as in a book chapter or a rather large section of a document (Akiva & Koppel, 2012). In such cases there is sufficient data for the algorithm to generate feature vectors to discriminate between the works of different authors. However, such studies ignore the contributions of other authors who might have written just a sentence or a paragraph within the document.

State of the art studies are based on reduced text length to identify the change in style in paragraphs, sentence groups, sentences or even a word. Studies focusing on determining style changes in a sentence or words are rare (Castro-Castro et al., 2020; Deibel & Lofflad, 2021; Safin & Kuznetsova, 2017). The fundamental task in writing style change detection can be considered as the task of separating single authored from multi-authored documents. It involves examining a document for possible style changes; the existence of style change signifies multiple authors while the lack of it indicates the presence of only one author (Kaur et al., 2020).

The other tasks of writing style change detection include finding positions in which authorship changes in a multi-authored document, determining the number of authors in a multi-authored document, and assigning each section of a document to an author. Solutions to these tasks have been systematically sought with increasing complexities. For instance, the first attempt to solve the problem of style change detection sought to determine whether a document is single authored or multi-authored, and for each multi-authored document, determine the position of authorship switches (Tschuggnall et al., 2017). The proposed approaches yielded poor results which did not meet defined performance baseline defined for the task, hence proving that it was a difficult task. In the following year, the style change detection task was broken down to the fundamental task of style change detection, and the preceding tasks thereafter defined with increasing complexities by combining two or more tasks; a previous successful task and a new more difficult task (Kestemont et al., 2018; Nath, 2019; Zangerle et al., 2019, 2020, 2021).

The style change detection methods are further categorized based on the main task it sought to solve as below:

I. Determining whether a document is single-authored or multi-authored

Different methods have been proposed by existing studies to solve this task. Some of these methods rely on the analysis of the different stylometric features to detect stylistic changes in a document (Hosseinia & Mukherjee, 2018; Rexha et al., 2018), while others adapted the outlier detection methods used in plagiarism detection problems. In addition, some studies investigated the use of hierarchical attention networks to solve this problem (Kestemont et al., 2018). Khan, (2018) used comparison algorithms on various stylometric features such as word frequencies of stop words and other POS words, punctuation, word pair frequencies and POS pair frequencies. The document is first segmented to various sentence groups and a stylometric match score calculated to check for style changes. The final document score is the sum of the various scores obtained from the sentence groups. This method yields good runtime although it does not produce good accuracy.

Hosseinia & Mukherjee, (2018) proposed a parallel hierarchical attention network to establish whether a document is multi-authored or not. In this approach, the feature set involved the parse tree features extracted from the tree-based structure of a sentence in order to preserve word order in a sentence. To determine style changes in documents, a fusion layer consisting of several similarity functions is used to compute the similarity/differences between the pair of documents. Specifically, they use the weighted vector and its reverse version in the comparison and to check for the existence of style changes in documents. While approached the style breach detection task by applying a sentence outlier detection commonly used in intrinsic plagiarism detection method. Although this approach achieves promising results, it took too long to run because of the PTFs whose production is very slow, especially on Stanford stand-alone parser.

II. Determining whether a document is single or multi-authored, and finding the borders where styles change

This task is an expansion of the fundamental style change detection task which examines a document to determine whether it is single or multi-authored. This task has been solved using various clustering algorithms; to separate single from multi-authored documents, and authorship linking which breaks down a document into smaller sections to establish whether there are authorship changes in the various sections. Literature defines a number of clustering algorithms for the complete authorship clustering and authorship linking; distance measures, B-compact graph-based clustering, compression-based clustering, hierarchical clustering algorithms and local sensitive hashing algorithms (Alberts, 2017; Adorno et al., 2019; Halvani & Graner, 2017).

Simple distance measures which clusters documents written by the same author together based on the distances between them have been used to solve the problem of complete author clustering and authorship linking. For complete author clustering, this method takes the absolute differences of any two vectors element-wise and sums them up to form summations which are used to check for writing style changes. The summations are transformed to standard deviations, where a high standard deviation score yields more evidence that the pair of documents is written by the same author. For instance, (Alberts, 2017) used SPATIUM-L1 on character n-grams to solve the problem of authorship clustering. They investigated with different character n-grams and achieved best performance at character 2-grams, with the top 300 most frequent features at threshold of 3.0 symmetrical score. Another study (Kocher & Savoy, 2017) used SPATIUM, on most frequent words, punctuation and character n-grams of each selected text. To measure the distance between a text A and another text B, they used a variant of SPATIUM; L-norm called Canberra in which the absolute differences of the individual features are normalized based on their sum.

The other approach that has been used to solve this problem is the B-Compact graph-based clustering. The method is based on defining a threshold function, which places documents into the various clusters only if the similarity between a pair of documents exceeds the threshold value. For instance, Garcia-Mondeja et al., (2017) proposed a method for discovering author groups using a -compact graph-based clustering. In this method each document is represented using the classic bag of words tried on different features. Similarity functions are then used to compare the similarity between a pair of documents, using only binary features. A threshold function is used to place documents into clusters only if the similarity between two pairs exceeds the threshold value of 0.5.

Compression-based algorithms have been proposed to solve the problem of complete author clustering and linking problems. Halvani & Graner (2017) used compression-based algorithms to perform document clustering into distinct clusters; they modified the K-Medoids algorithm using a compression-based dissimilarity measure as opposed to the standard distance measure. The value of k- which represents the number of authors was determined by computing silhouettes coefficients in an iterative manner. N-clustering iterations were performed and the value of k that produced the maximum silhouette coefficient was picked. For the authorship link, they applied a dissimilarity function, compression-based cosine to measure how dis-similar two documents are to each other. In order to establish authorship links within each cluster, compression-based cosine was modified to calculate similarity score instead of dissimilarity score. This approach does not perform well because compression-based dissimilarity measures do not fulfill even one of the required properties of a real distance-based metric such as identity, symmetry and triangle inequality.

Adorno et al., (2018) proposed a hierarchical clustering analysis of different document features: typed and un-typed character n-grams and word n-grams for the complete author clustering. Hierarchical clustering analysis was used to determine the number of distinct clusters and to place each document in exactly one of the k-clusters. The hierarchical analysis was done using the bottom-up approach where each text starts in its own cluster and after each iteration, a pair of clusters are merged. The average cosine distance is used to decide when to merge pairs. To establish authorship links, pairwise similarity between each pair of documents in each problem was calculated using the cosine similarity metric. The use of the same feature set for all languages may have had a negative effect on the overall performance. Using different features for each language may help improve the problem.

Karas et al., (2017) proposed a method for author clustering and style breach detection based on local-sensitive hashing-based clustering of real-valued vectors; a mixture of stylometric features and bag of n-grams. TF-IDF features and the Wilcoxon signed rank test were computed to determine the style breaches. The study investigated two Local-sensitive hashing algorithms;

super bit and min Hash and found out that super bit, which approximates cosine similarity, yielded the best results in author clustering. Silhouette coefficient was computed to determine the number of clusters. For the style breach detection, a statistical approach- Wilcox-on signed Rank, based on TF-IDF features was used to determine the borders of the changing styles within a document.

These approaches used unsupervised techniques and therefore are applicable to solving real-world problems where the number of participating authors is not known in advance. Better still, they employ very simple techniques; distance measures and other clustering algorithms on standalone features thereby yield low run times. However, the results posted by these methods are slightly above the baseline and still require strengthening. Expanding feature sets could greatly improve the performance of these methods since features are generated at the sentence level, therefore just one feature type may not adequately represent the writing style of an author.

III. Is a document multi-authored, if yes determine the number of authors who collaborated?

Determining the number of authors in a multi-authored document is the goal of style change detection. However, the need to subject the model only to multi-authored documents necessitates the separation of single authored from multi-authored documents (Zangerle et al., 2019). Whereas this task can be inherent in a model for determining the number of authors in a document, it is essentially done first to minimize the number of documents passing through the algorithm for predicting the number of authors based on similarities in styles of writing and to improve algorithm efficiency (Deibel & Lofflad, 2021; Sari & Stevenson, 2016). Supervised and unsupervised learning techniques have been used to tackle this problem.

Akiva & Koppel, (2012) used unsupervised learning to determine the number of authors in a multi-authored document using 500 most common words as the style marker. They defined two levels, firstly to cluster the chunks into two, three or four author clusters, using cosine similarity. They then applied supervised learning on an expanded feature set to distinguish between the clusters. They found out that unsupervised learning yielded better results than supervised learning, however as author numbers increased, the algorithm accuracy reduced.

Nath, (2019) used a combination of features to establish the number of writers in documents. They defined an algorithm using an ensemble of two unsupervised learning algorithms; a threshold based and window merge clustering methods. This study first employed the threshold algorithm to cluster windows based on their closeness. That is, windows with the smallest distances between them, are put in one cluster because it is assumed that such windows have the same author. Then the most similar windows were merged using the distance matrix to calculate the distance between the new windows. The study found out that Threshold Based Clustering outperformed the Window Merge Clustering. Although the use of duplicate sentences improved significantly the accuracy, it also led to an increase in the OCI value.

Zuo, Zhao & Banerjee, (2019) Defined a two-pipeline for determining style changes in documents. First, they used a feed forward neural network to categorize single authored documents from multi-authored documents. They then applied a 3-algorithm clustering to establish the number of writers in the multi-authored texts. To cluster segments into groups in a multi-authored document, they used various combinations of stylometric features and an ensemble of clustering algorithms. The ensemble consisted of k-means, k-means with similarity and hierarchical clustering. K-means clustering algorithm was used to separate single-authored documents from multi-authored ones. To form the clusters, they employed silhouetting on the k-means algorithm to determine the number of clusters. To establish the number of writers in a document, hierarchical clustering was used on all the features except the TF-IDF features, together with the feed forward neural network to determine the exact number of clusters in multiple authored documents. The study noted that classification results varied with an increasing number of authors in a document.

IV. Is a given document multi-authored, if yes is there a style change between consecutive paragraphs?

Determining the change in style between consecutive paragraphs can be approached as a supervised learning problem by generating feature vectors for each paragraph and comparing these feature values (Zangerle et al., 2020). It can be solved using paragraph representations or simply by breaking the document into sentences and generating features at the sentence level (Castro-Castro et al., 2020; Lyer and Vosoughi (2020). For instance, Castro-Castro et al., (2020) used characters, lexical and syntactic style markers to build a paragraph representation to establish the number of writers of a document and the corresponding paragraphs authored by each. The study grouped Paragraphs according to a defined heuristic based on the B0- maximal clustering algorithm. This approach suffers from paragraph overlap. This problem was partly eliminated by considering the order of the paragraph in the document. This method assumes that the writing style in a document is characterized by the style reflected in the first paragraph, and that the main author tends to write the majority of the paragraphs, particularly the first ones. Whereas this assumption may be true the effects of other characteristics such as the size, strength of similarity or the adjacency of the paragraphs ought to have been considered. Paragraph overlap was a challenge with this approach.

The approach of Lyer and Vosoughi (2020) was based on using Google's BERT language algorithm as a feature extractor, and random forests as a classifier. First, the documents contained in the data set are split into sentences, and every sentence is fed to BERT, taking the outputs of the last four BERT layers to represent a given sentence. Since the size of the feature matrix produced by this depends on the number of tokens in a sentence, the values along the length dimension are summed to obtain a feature matrix of a fixed length. After this, representations are formulated for consecutive pairs of paragraphs (to solve the second task), and the whole document (to solve the first task), based on the representations of sentences, by summing (paragraphs) or averaging (whole documents) the feature values of the sentences that make up the paragraph or document. These feature representations are then used to train random forest algorithms for both tasks. Although this method posted the best results for this task, it did not use stylometric features which this study believed could also compete the BERT algorithms.

V. Given a text, find out whether the text is written by a single author or by multiple authors. For each multi-authored text, find the positions of the changes and assign all paragraphs of the text uniquely to some author out of the number of authors you assume for the multi-author document.

This task combines all the other tasks of style change; author clustering, authorship linking, number of authors and finally introduces a new task of assigning paragraphs of the text uniquely to an author. It has been approached using supervised learning techniques, which yield better accuracy, by performing pairwise comparisons of paragraphs. For instance, Deibel and Lofflad (2021) proposed the use of multi-layer perceptron and bidirectional LSTM for the style change detection. Widely used textual features such as mean sentence length in words, mean word length or corrected type-token ratio, and pre-trained Fast-text embedding. Multi-layer perceptron with three hidden layers that are fully connected are used to categorize single authored texts from multi-authored text.

Kaur et al., (2020) trained a logistic regression classifier on the absolute vector difference between the feature vectors corresponding to each paragraph pair to solve the problem of style change detection. If the average of the classifier scores corresponding to the adjacent paragraph pairs is greater than 0.5, then the document is multi-authored.

Strom, (2021) used a stacking ensemble of classifiers trained on separately extracted features and BERT embedding, and combined their predictions by a meta-learner, i.e the stacking ensemble. Classifying single or multi-author documents was achieved by classification on the document level features. A single feature vector per document and label was used to classify each document as being either single or multi-authored.

Zhang et al., (2021) used google's pre-trained BERT algorithm to determine the style change detection as a binary classification problem based on the similarity of writing style. They modeled the problem of writing style change detection as discovering the similarity of writing styles between different text segments. 'The style changes and the decision of author identifiers were regarded as binary'. They adopted the Bert pr-training algorithm to extract the paragraph features and build a algorithm to solve all the style change detection problems outlined in the competition. In this algorithm they report that if task 2 label includes 1, the corresponding text will at least be two authors, and the corresponding task1 label will be 1. Otherwise task 1 label will be 0. Two paragraphs were presented for similarity measurement where high similarity indicates no change in writing style between the two paragraphs. A low similarity denotes change in writing style. To estimate writing style similarity, BERT together with Fully connected Neural network classifiers were used.

2.2.3 Critique of Related Studies on Writing Style Change Detection

Few studies focus on writing style change detection; a case of determining the number of authors or the number of style changes in multi-authored documents. The overview of existing writing style change detection models is provided in table 2.1.

Table 2.1: An overview of existing writing style change detection models

SN	Features	Algorithms	Data	Weakness	Reference	How the weakness will be solved
1.	Combination of features.	Ensemble of Threshold and Window Merge Clustering	PAN 2019	Defined writing style change decision at the paragraph level.	Nath, (2019)	Defining the writing style change at the sentence level and ensembles of clustering algorithms
2.	Bag of words and combination of features	K-means clustering, Hierarchical clustering, and Feed forward Neural Network	PAN CLEF, 2019	Defined writing style changes at the paragraph.	Zuo, Zhao and Banerjee, (2019)	Defining writing style changes at the sentence level
3.	Most common words	Unsupervised learning	Own data	Known number of authors and same topic scenario and a Small feature set	Koppel and Akiva, 2012	Use ensembles of clustering algorithm and optimal feature sets
4.	Characters, lexical and syntactic	Paragraph representation using B0-maximal clustering	PAN CLEF, 2019	Paragraph overlap	Castro et al., 2019	Use ensembles of clustering algorithm and optimal feature sets

Table 2.1 presents the existing models considered most related to the study. Here only studies that focused on determining the number of authors in multi-authored document were analyzed. For instance, Zuo, Zhao and Banerjee, (2019) defined a two-step pipeline for determining style changes in documents. First, they used a feed forward neural network to categorize single authored documents from multi-authored documents. They then applied a 3-algorithm clustering to establish the number of writers in the multi-authored texts. To cluster segments into groups in a multi-authored document, they used various combinations of stylometric features and an ensemble of clustering algorithms. The ensemble consisted of k-means, k-means with similarity and hierarchical clustering. K-means clustering algorithm was used to separate single-authored

documents from multi-authored ones. To form the clusters, they employed silhouetting on the k-means algorithm to determine the number of clusters. To establish the number of writers in a document, hierarchical clustering was used on all the features except the TF-IDF features, together with the feed forward neural network to determine the exact number of clusters in multiple authored documents. This approach obtained an accuracy of 0.6 for the binary classifier, while the multi-author detection model realized an Ordinal Classification Index (OCI) of 0.808 which was slightly better than the average. In this study writing style changes were defined at the paragraph level resulting in a possibility of the model ignoring changes that happen at the sentence level.

Akiva & Koppel, (2012) used unsupervised learning to determine the authors in a multi-authored document using 500 most common words as the style marker. They defined two levels, firstly to cluster the chunks into two, three or four author clusters, using cosine similarity. They then applied supervised learning on expanded feature set to distinguish between the clusters. This approach yielded accuracy of 88%-96% for author pairs, for documents with three authors this method obtained an accuracy of 77%-82% while a purity of 74% was obtained for documents with four authors. They found out that unsupervised learning yielded better results than supervised learning, however as author numbers increased, the model accuracy reduced. In addition, in this method participating authors' number is known, simplifying the approach further. Moreover, the study did not use lexical features which have been shown to yield better accuracy in style change detection. Potential limitations of this study were; the method is only applicable in when the said documents fall in the same topic and genre. Otherwise it is difficult to distinguish same-author and different-authors pairs. Lack of certainty on the authorship of the impostor document was also a major limitation in this study, as these impostor documents could as well have the same authors as of the pair of documents being investigated.

Castro-Castro et al., (2020) used characters, lexical and syntactic style markers to build a paragraph representation to establish the number of writers of a document and the corresponding paragraphs authored by each. The study grouped Paragraphs according to a defined heuristic based on the B0- maximal clustering algorithm. This approach obtained an average F1score of 0.6489. Despite coming second in the PAN 2020, this approach suffers from paragraph overlap. This problem was partly eliminated by considering the order of the paragraph in the document.

This method assumed that the writing style in a document is characterized by the style reflected in the first paragraph, and that the main author tends to write majority of the paragraphs, particularly the first ones. Whereas this assumption may be true the effects of other characteristics such as the size, strength of similarity or the adjacency of the paragraphs ought to have been considered. Paragraph overlap still remains a challenge.

Another study Nath, (2019) used a combination of features to establish the number of writers in documents. They defined an algorithm using an ensemble of two unsupervised learning algorithms; a threshold based and window merge clustering methods. This study first employed the threshold algorithm to cluster windows based on their closeness. That is windows with the smallest distances between them, are put in one cluster because it is assumed that such windows have the same author. Then the most similar windows were merged using the distance matrix to calculate the between the new windows. The study found out that Threshold Based Clustering outperformed the Window Merge Clustering. This method obtained best performance at an accuracy of 0.6, an OCI of 0.82 and a final rank of 0.42. However, when duplicate sentences were used as a feature, an accuracy of 0.85, OCI of 0.87 and a final rank of 0.49 were achieved. Besides achieving top performance, the final rank did reach the mean of 0.5. Although the use of duplicate sentences improved significantly the accuracy, it also led to an increase in the OCI value. Better still, this scenario was unique to this data set and may not provide a good generalization.

So far, the performance of the existing models on style change detection is not competitive, compared to the other models in other authorship verification tasks. Few models have managed to attain the mean mark (0.5) in the overall performance. However, most models report accuracy slightly above the baseline, 0.5. This study aimed to achieve an upper bound accuracy of 0.8 and OCI of 0.8 and a final rank r, of 0.5. Such levels of accuracy are considered good enough for authorship verification models (Amigud et al., 2017; Brocardo et al., 2013; Zangerle et al., 2019). An overview of performances of the existing models on writing style change detection is given in table 2.2.

Table 2.2: Performance of Existing Models on Writing Style Change Detection

Reference	data set	Accuracy	OCI	F1_Score	Rank
Akiva and Koppel, (2012)	Own	0.74			
Castro-Castro et al., (2020)	PAN 2019			0.65	
Nath, (2019)	PAN 2019	0.85	0.87		0.49
Zuo, Zhao and Banerjee, (2019)	PAN 2019	0.6	0.80		0.40

2.3 Review of the Performance of Base Algorithms

Different machine learning algorithms have been used for the task of writing style change detection. For the supervised machine learning category, a number of algorithms exist which can be used to detect writing style changes in documents. Some of the commonly used supervised learners are Naive Bayes, Support Vector Machines, Logistic Regression, k-Nearest Neighbor, Random Forest, Decision Tress among others. The performance of these base algorithms vary depending on the task at hand, however some studies have evaluated and compared the performance of these algorithms on text classification and sentiment analysis.

For instance, a study by Pranckevičius and Marcinkevičius, (2017) sought to compare performance of common classifiers such as Logistic Regression, SVM, Naive Bayes, Random Forest and Decision Trees, found out that Logistic Regression achieved the highest classification accuracy compared to SVM, Naive Bayes, Random Forest and Decision Trees. Another study by Shah et al., (2020) conducted a comparative analysis of Random Forest, Logistic Regression and k-NN on text classification. The study experimented with different data sets and reports that Logistic Regression achieved the highest average classification accuracy at 0.97, followed by Random Forest at 0.93. K-NN achieved the lowest accuracy at 0.92.

A study by Guia et al., (2019) sought to compare performances of four common classifiers on sentiment analysis. The study compared the performance of Random forest, SVM, Naive Bayes and Decision Trees. It was reported that SVM achieved the highest performance values in terms of accuracy, precision, recall and F1score. Specifically, SVM achieved an accuracy of 0.89 followed by random Forest at 0.87. to this end, it can be seen that Logistic Regression, SVM and Random Forest are among the best base algorithms for text classification.

Although Logistic Regression is among the best classifier, few studies have used in writing style change detection. Among the studies are the work by Kaur et al., (2020), which trained a logistic regression classifier on the absolute vector difference between the feature vectors corresponding to each paragraph pair to solve the problem of style change detection. If the average of the classifier scores corresponding to the adjacent paragraph pairs is greater than 0.5, then the document is multi-authored.

Unsupervised learning techniques have also been employed in writing style change detection studies. The most commonly used of which are the K-means, hierarchical and probabilistic clustering methods. K-means is not only simple to implement, but also yields better performance in terms of accuracy compared to other clustering algorithms. For instance, Singh and Singh, (2012) compared the performance of K-means clustering algorithm and Hierarchical clustering algorithms in terms of accuracy and run-time. They report that K-means achieved highest performance level at an accuracy of 0.89 compared to hierarchical at an accuracy of 0.66 on the Iris data set. Based on the run-times, K-means outperforms hierarchical clustering achieving a maximum runtime of 0.03, compared the Hierarchical which achieved a runtime of 0.17.

Another study compared the performance of K-means clustering and Expectation Maximization methods of Gaussian Mixture Models in terms of accuracy and run-time. This study concludes that K-means achieves more accurate clusters compared to GMMs, but at a slower speed (Jung, Kang & Heo, 2014). Clustering algorithms differ in their performance abilities such as run-time and accuracy. However, it can be deduced that although k-means has slower run-times compared to GMMs, it remains a powerful clustering algorithm for most tasks involving text. The difference in terms of runtime between K-means and GMMs is not very significant if the accuracy of clusters is the main focus. Hierarchical Models are slower in terms of runtime compared to k-means and GMMs.

Ahmadi, (2008) compared performance of four clustering algorithms; K-means, GMMs, Hierarchical and Self Organizing Maps (SOMs). Performance evaluation was based on a number of factors such as data set size, number of clusters, type of data set and type of software used. This study notes that K-means and GMMs are similar in most cases but differ in how the distance measure is calculated. Therefore the study reports similar observations with both K-means and GMMs and a different similar trend with SOMs and Hierarchical algorithms. In general, it is noted that Hierarchical algorithms and SOMs achieve better results in terms of accuracy in small data sets compared to K-means and GMMs. Moreover, when the number of clusters (value of k) is increased, better performance is observed with K-means and GMMs until the performance converge with the performance of SOMs and Hierarchical clustering algorithms.

It evident that the choice of which clustering algorithm to use therefore depends on a number of factors such as data set size, number of clusters and type of data set. However, the use ensemble of different clustering algorithms in a bagging fashion could yield better performance by maximizing on the strength of each individual clustering algorithms.

Ensemble Learning

Ensembles, which yield better performance compared to individual algorithms were considered in the design of the two models. Bagging ensembles are used to optimize the performance of the model since its results always outperforms the results of a single algorithm. In addition, a bagged ensemble is more resilient to noise and does not easily over-fit on a data set as is with boosting ensembles. Ensemble learning was used in this study because the base algorithms complement each other thereby providing better performance. In addition, they have yielded promising results in previous studies (Oloo et al., 2022a).

For example, Zlatkova et al.,(2018)'s stacking ensemble classifier was the most effective method for separating single-authored texts from multi-authored documents. When it came to determining the number of style changes within a document, Zuo, Zhao and Banerjee, (2019)'s ensemble clustering approach fared better than previous research (Zangerle et al., 2019). When there is no prior information of the authorship, clustering approaches have demonstrated their effectiveness in identifying latent stylistic tendencies that can be used to discriminate between numerous authors (Alshamasi & Menai, 2022).

Ensemble models are better performers since they have reduced biases and improved variance which ensures that results can be generalized. The main challenge in ensembles learning lies in the choice of the number of base algorithms to include in the ensemble and the base algorithms to use.

2.4 Determining Optimal Feature Sets for Writing Style Change Detection

The third objective of the study was to determine the optimal feature set for detecting writing style changes in multi-authored documents involving short text length. The design of stylometry-based writing style change detection models is influenced mainly by the stylometric features used and the machine learning model, (Amigud et al., 2017). Literature provides different model designs distinguishable by the nature of the stylometric features and machine learning algorithms.

Whereas a rich set of stylometric features exists applicable to different tasks of the writing style change detection, there lacks a consensus on which is the best set of features that yields optimum performance for every task (Anwar, 2019). In addition, the distribution and usefulness of features differ for different data sets. Therefore, determining the right set of features to use is very important for the task of writing style change detection (Kestemont et al., 2018).

2.4.1 Categories of Stylometric Features

Writing styles have been defined by stylistic elements, which look for traits that an author uses consistently in all of their works (Brocardo et al., 2013, 2015; Juola, 2006). It has been possible to identify whether a document is authored by a well-known author, whether two or more documents share the same author, or whether various authors contributed to different sections of the same document through the analysis of writing style similarities. Studies on authorship verification and writing style change detection have made use of a wide range of stylometric features. While new features are continually being developed, Brocardo et al., (2013) report the existence of over a thousand stylometric features grouped in five categories. Some of the examples of these features are function terms, POS terms, lexical terms like word n-grams, type-token ratios, word length, misspellings, character n-grams, among others.

Previous research classifies stylometric features into three, four, or five groups based on the nature of the job. Syntactic, content-specific, and word-based traits are the three categories Gelbukh (2015) uses to group stylometric features. Four categories of stylometric features are defined by Abbasi and Chen (2006): lexical, syntactic, structural, and content-specific features. Brocardo et al. (2015) divides lexical features into two categories: lexical and character features, giving rise to five categories: lexical, character, syntactic, structural and content-specific features. This allows for the classification of stylometric aspects into five groups. The list of categories of stylometric features utilized in writing style change detection is extensive and can be found below.

2.4.1.1 Lexical Features

Lexical features can be used to infer a writer's preference for using particular words. Some examples of lexical features such as most frequent terms, adverbs, nouns, pronouns adjectives, word form errors etc. Verbs are action words in a sentence which can be used to indicate someone's action or express a state of being. Similarly, adverbs are used to describe a verb or to show how the action expressed by the verb was carried out. Nouns are words used to identify a place, object, person or idea. Adjectives are parts of speech words which can be used to describe or provide more information about a noun or pronouns such as tall, short green red etc. The choice and use of these words vary from one author to the other. For instance, the number of occurrences of each lexical category can be different for each author and can therefore be used as a feature to differentiate between works of different authors.

Tokenization, a technique that divides a text into tokens representing each of its individual words, can be used to extract these features. They can be extracted both at word or sentence level. Lexical features at the word level might comprise, among other things, word uni-grams, stop words, POS words, word frequencies, word n-grams, and vocabulary richness. Features at the word level have also been derived from word level statistics, including word length, total number of words, average word length, most frequent words, and type-token ratio.

Word-level features have been used for a variety of writing style change detection tasks, including author clustering, differentiating between single and multi-authored texts, counting the number of style changes in documents, and pinpointing the locations of style changes (Zangerle et al., 2019, 2022). For instance, in order to calculate paragraph similarities for the purpose of identifying style changes, Nath, (2021) employed a Siamese Neural Network on vocabulary richness. When dealing with short text, like sentences, word-level characteristics could be a suitable option for detecting writing style changes.

Studies have employed many features, including sentence length, average sentence length, misspellings, and repetitions, as style markers at the sentence level. When the document length is significantly longer, as in sentence groups, paragraphs, or even full texts, these features produce promising results (Adorno et al., 2017; Karas et al., 2017; Kuznetsov et al., 2016; Safin & Kuznetsov, 2017). Nath (2019) produced highly encouraging results for the writing style change detection tasks by combining a number of features. Moreover, the study found that using duplicate sentences significantly improved performance. For the style change detection task, Deibel and Lofflad (2021) used pre-trained Fast Text embedding with multi-layer perceptron and bidirectional Long Short Term Memory, as well as mean sentence length in words, mean word length, or corrected type-token ratio.

Based on previous research, it can be concluded that lexical features are most frequently employed in writing style change detection investigations. These features are very popular since they may be employed in studies based on short texts and can be used without incurring additional costs across languages. Lexical features are used in most prior studies because they offer a solid gauge of the measurable stylistic distinctions within a writing style (Brocardo et al., 2015; Castro-Castro et al., 2020; Ding et al., 2016; Howedi et al., 2020).In addition to general authorship verification problems, this feature category continues to yield promising results in other specialized tasks like authorship clustering, writing style change detection, and writing style change over time (Kocher & Savoy, 2017; Kestemont et al., 2018; Lyer & Vosoughi, 2020; Strom, 2021).

Although lexical features are the most often utilized features in writing style change detection, there is still debate regarding the purity of algorithms that solely rely on these features due to their topic dependence and potential to introduce topic, genre, and domain effects (Abbasi & Chen, 2005; Juola, 2006; Rosso et al., 2016).

2.4.1.2 Character Features

Disparities in the use of lexical information, such as capitalization and punctuation, are captured by character features (Brocardo et al., 2013; Sari, 2018). Among these are character n-grams, the total amount of characters, the total number of numbers, the total number of capital letters, the total number of space characters, the number of tabs and their corresponding ratios, and the use of special characters. Tokenizing words into characters allows for the extraction of character features. Character features have been used in previous research either alone or in an ensemble. For example, Rexha et al. (2018) examined the impact of an ensemble of supervised learning algorithms and n-grams. Character 2-grams perform best when used with the top 300 most frequent features, according to Alberts' (2017), who investigated the use of various character n-grams. Using common n-gram profiles of text documents from the PAN 13 data set, Jankowska et al. (2017) looked at author verification in another study.

A few studies combined character features with additional features to create an ensemble of features. For example, Karas et al. (2017) suggested an approach based on local-sensitive hashing-based clustering employing a bag of n-grams and other stylometric variables for author clustering and style breach detection. Character statistics, such as the most common punctuation, n-grams, special character frequencies, etc., or character frequencies themselves have also been employed. Kocher and Savoy (2017) employed the most frequent terms (separated words and punctuation symbols) and most frequent character n-grams of each text to apply a straightforward unsupervised author clustering and authorship linkage technique named SPATIUM. Because these features are resistant to noise from typos and misspellings, they are regarded as superior style markers (Sari, 2018).

While it is possible to extract character features from very short texts, like a sentence or even a word, research indicates that these features are insufficient to adequately capture stylistic differences in short documents (Karas et al., 2017; Kestemont et al., 2018; Nath, 2019; Strom, 2021). In order to improve writing style change detection accuracy, an ensemble of these features along with other features may be built.

2.4.1.3 Syntactic Features

The only reliable way to compare styles of writings by the same or different authors is through syntactic features. They can be quantified and normalized, which makes them a lot easier approach to express writing styles (Brocardo et al., 2013; 2015). The most often utilized syntactic features are punctuation and common words, like determiners, prepositions, conjunctions, and pronouns. Stop words, which make up the majority of words in any text, are also frequently used as features. Punctuation refers to the system of symbols author uses to make the meaning of their sentences or work clear. They include period, comma, question mark, exclamation mark, semi-colon. In other words, all the tools an author uses to separate sentences, phrases and clauses so that the meaning is clear.

Frequencies of both functional and part of speech words and frequencies of punctuation can also be used as style markers. They are obtained by simply counting the number of punctuation or part of speech words. Using stop words and other stylometric features, Karas et al. (2017) presented a method for author clustering and style breach detection based on local-sensitive hashing-based clustering. In order to create a comparison algorithm for identifying stylistic changes inside a document, Khan (2018) employed stop words and other fundamental stylometric features. Certain studies have also made use of function terms. To determine the number of writers of a document and the related paragraphs produced by each, Castro-Castro et al. (2020), for example, built a paragraph representation using characters, lexical, and syntactic style markers.

Syntactic features have typically been utilized in conjunction with other features to identify stylistic differences in documents in prior research. Because they need a syntactic parser to understand particular natural languages, they are costly even if they are thought to offer the finest authorial signature (Alberts, 2017; Brocardo et al., 2013; Rosso et al., 2016).

2.4.1.4 Structural Features

Based on how each author arranges their work, structural features can be utilized to validate stylistic variations in documents written by several writers (Juola, 2008; Brocardo et al., 2013). The number of sentences in a paragraph or text, the average word count, the average character count, the average sentence length, and the average number of sentences that start with upper- and lower-case letters are a few examples of structural aspects. Document, section, and technical level are examples of structural features that can be defined at three different granularity (Adorno et al., 2017; Gelbukh, 2015).

Font style, font color, use of tab spaces, paragraph alignments, spacing style, length of sentences and length of words are features which define the general organization of a document. The preference to certain font style or font coloration can be used to model an author’s style. Similarly, the choice of the various spacing styles or the general preference to a specific spacing style can indicate works of a specific author. Length of sentences which can be measured by the total number of words or characters of a sentence is also a common style marker in authorship analysis studies. For instance, some authors use short sentences while others prefer longer sentences. Therefore, documents with consistently longer sentences could indicate one style while where shorter sentences are used could indicate change in writing style.

Readability, which gauges how simple a text is to read, is categorized under structural features at the technical level. Previous studies have employed a variety of readability metrics and kinds, including the Automated Readability, Dale-Chall readability score, and Simple Measure Of Gobbledygook (SMOG) index. Modern writing style change detection investigations have successfully employed readability criteria, despite the fact that they are not frequently utilized in conventional authorship attribution studies (Zlatkova et al., 2018; Zuo, Zhao & Banerjee, 2019). Because of their cumulative influence, structural features might be a useful choice for writing style change detection investigations involving short text lengths, such sentences, where other features would not be able to discriminate between two styles sufficiently.

2.4.1.5 Context Features

Context features are those that indicate the presence of specific keywords, interest groups, and activities. They offer precise details and contextual information relevant to the current job. The most frequently used context features are frequency of emoticons, forwards and tagging, number of links to other pages and hash tags. Emoticons are a characterization of a facial expression such as smile or frown, found by various combinations of keyboard characters and used to show the author’s feelings or expected tone. They are commonly used in social networking environments whereby people use them to convey their feelings. Some people use so many emoticons while others limit their use to very few or to none altogether. Therefore, counting frequency of use of emoticons can be used to differentiate between styles of authors. Hash tags are a way to attach social media content to a specific topic, event, theme, or conversation. They are words or phrases preceded by the pound (#) symbol.

In the context of online sales settings, for example, Juola (2006) found several essential phrases like obbo, windows, hashtags, etc. by carefully observing and analyzing studies in ancient times (Abbasi & Chen, 2005; Kaur et al., 2020; Sittar et al., 2016). Seldom do studies employ content-based features to identify variations in writing styles within publications because they are not significant stylistics measures in cases where different topics are used. The identification of authors for documents with strong content similarity was investigated by Rexha et al. (2018).

Their study concentrated on examining how readers evaluate various writing styles according to writers' styles that are independent of content.

2.4.2 Feature Selection

A crucial task for the machine learning-based writing style change detection investigations is feature selection. According to Brocardo et al. (2013), Nath (2019), and Zlatkova et al. (2018), the effectiveness of these algorithms depends on selecting an appropriate feature set that may simulate an author's writing style. This task holds significance as it mitigates overfitting on the data set, enhances the purity of the style change detection method, and lowers computing expenses. The current state of feature engineering techniques can be divided into three groups: experimental approaches, manual feature selection, and feature reviews (Nath, 2019; Pandian et al., 2020; Zuo, Zhao & Banerjee, 2019).

2.4.2.1 Manual Feature Selection Method

The most popular technique for choosing features in early authorship investigations was the manual feature selection method. Using this approach, the data set is manually examined for characteristics that can distinguish between various styles. To ascertain the number of authors in multi-authored documents, four studies employed the use of manual feature selection methods. For instance, Koppel and Akiva (2012) suggested an approach based on most frequent terms. For the authorship verification research on short text, Juola, (2006) and Brocardo et al. (2013) manually selected features.

Although it is a laborious process and may not be possible for features that are manually created to deliver better performance in shorter documents such as sentences, for example, it is generally a more efficient way to produce features. Very promising outcomes have been obtained from a small number of features that were manually built by searching data sets for decisive features. Nath, (2019), for example, used a collection of features that have been shown to yield positive results in other studies to assess the number of authors in a text. Finding the number of authors in multi-authored documents was suggested by the study using an ensemble of three clustering methods. When duplicate sentences, a feature found through document analysis were employed, algorithm performance improved.

2.4.2.2 Feature Reviews

The majority of recent studies on writing style change identification has chosen the features for their suggested techniques based on features and feature sets that yielded encouraging findings in earlier related studies. This method chooses features, or a subset of features, that perform better on the same or related tasks than the rest, either comprehensively or expanded. For example, Zuo, Zhao and Banerjee. (2019) counted the number of authors in multi-authored documents using a feature set derived from the PAN competition winning submission from the previous year. In contrast, Nath (2021) created a technique based on Google's BERT embedding, which has been demonstrated to yield the greatest results in experiments on style change identification.

2.4.2.3 Experimental Feature Selection

Experimental feature selection techniques including chi-square, information gain, frequencies, and entropy, among others, have not been widely used in research. Computing frequencies is the most widely used method in previous studies to engineer features partly because the most frequent features in a data set may also be the most significant in distinguishing between writing styles in documents. Gorman (2020) tried to figure out which features would be most useful for a categorization task. With the expectation that these features will be the most important for this assignment, they extracted the most common feature categories. Their strategy used a ranking of feature types by frequency to choose the best feature set to use. They reported that the most significant feature types were those at the top of the list.

Pandian et al., (2020) to choose features for the Tamil language. They conducted experiments with decision trees, C4.5 algorithms, Support Vector Machines, and Classification Based Associations (CBA) in order to validate their suggested strategy. Compared to features chosen by other methods, they reported better performance when decision trees were employed to pick them.

Features have also been chosen by ranking them according to feature importance scores. According to this method, the features with the highest scores are the most significant, and vice versa. Gorman, (2020), for example, employed feature ranking by frequency to determine which features would be most useful in a classification challenge. The most important features for the assignment were the top features.

When utilized for feature engineering, tree-based techniques like decision trees and random forests produce promising results. These classifiers are quick to train, evaluate, and interrupt. They are also non-parametric, which means that outliers have no effect on them. Their primary drawback is their propensity for over fitting, but ensemble techniques like Random Forest can mitigate this (Gorman, 2020). Compared to decision trees, are simpler to understand and train more quickly. For cross-genre and cross-topic data sets published in the four languages of Dutch, English, Greek, and Spanish.

Ghosh et al., (2016) suggested an automated authorship verification method. To select features, they used a random forest classifier and seventeen features including parts-of-speech (POS), vocabulary, sentence length, punctuation, and n-grams. The total number of punctuation, the ratio of specific punctuation, the ratio of long to short sentences, the strength of the vocabulary, the n-gram difference, the starting and ending POS frequencies, and the total number of punctuation.

Ablation studies which are frequently employed in the medical field have been performed to calculate the contributions of each feature category in various data sets. On several data sets, Sari et al., (2018) aimed to ascertain the contributions of three feature groups: style features, content features, and hybrid features. Four popular data sets for authorship attribution: CAT 10, CAT 50, JUDGEMENT, and IMDb62were employed in their experiments. A Feed Forward Neural Network and logistic regression were used to analyze the first 100 common n-grams. The results of the study showed that style-based features performed better compared to structural and content based features on data sets where writers covered comparable topics. On the other hand, data sets with dissimilar subjects demonstrated the value of content features. Because ablation tests compute feature importance, they are the most accurate feature selection methods.

2.4.3 Stylometric Features used in Writing Style Change Detection

The focus here was to identify features which had been applied in writing style change detection and authorship verification studies involving short length documents were identified and tabulated. The criteria of identifying the the works from which to draw the features was limited to studies focusing on writing style change detection, and authorship verification studies involving short text. We defined short text to mean documents with not more than five hundred word (500), adopting this definition from Brocardo et al., (2013). In this regard, a total of thirty nine studies were identified and the features which had been used in these studies tabulated in table 2.3.

Table 2.3: Most commonly used features in writing style change detection adopted from Oloo V. et al., (2022a)

CATEGORY	FEATURES	REFERENCES
Lexical	**Word Level features**	
	Word n-gram	(Ding et al., 2016; Gómez-Adorno et al., 2017; Gorman, 2020)
	Word frequencies	(Gorman, 2020; Khan, 2018)
	Vocabulary richness	(Karaś et al., 2017)
	Stop words count	(Alshamasi & Menai, 2022; Khan, 2018)
	Number of difficult words	(Zlatkova et al., 2018)
	Word length, total number of words	(Castro-Castro et al., 2020; Kaur et al., 2020)
	Average word length	(Alshamasi & Menai, 2022; Karaś et al., 2017; Alshamasi & Menai, 2022)
	most frequent words	(Alberts, 2017)
	Average word syllable	(Alshamasi & Menai, 2022)
	Word pair frequencies	(Kocher, 2016)
	Type_token ratio	(Deibel & Löfflad, 2021)
	Duplicate words	(Zlatkova et al., 2018; Zuo, Zhao & Banerjee, 2019)
	Most frequent terms	(Gómez-Adorno et al., 2017)
	Sentence Level features	
	Duplicate sentences	(Nath, 2019)
	Sentence length	(Alshamasi & Menai, 2022)
	Number of sentences starting with lower case letters	(Alberts, 2017)

	Total number of all-uppercase words in a sentence, Number of sentences starting with capital letters	(Castro-Castro et al., 2020; Kaur et al., 2020)
	Total number of miss-pelt words	(Zuo, Zhao & Banerjee, 2019)
	Total number of words in a sentence	(Alshamasi & Menai, 2022)
Character Level	Special characters such as , Digits, Alphabets, White spaces, Emojis	(Sittar et al., 2016)
	Character n-grams	(Alberts, 2017; Gómez-Adorno et al., 2017)
	n-gram count	(Karaś et al., 2017; Sittar et al., 2016)
	Tabs count	(Sittar et al., 2016)
	Special character frequencies	(Karaś et al., 2017)
	Total number of uppercase letter	(Sittar et al., 2016)
	Character frequencies	(Kocher, 2016)
	Total number of special characters	(Karaś et al., 2017)
	Most frequent character n-grams	(Gómez-Adorno et al., 2017)
	First word uppercase	(Sittar et al., 2016)
Syntactic Features	Punctuations such as single quotes, commas, periods, colons, semi-colons, question marks, exclamation marks, and special marks based on Unicode format.	(Kocher, 2016)

	Part of Speech words (POS) including common words such as pronouns, prepositions, adjectives, interjections, conjunctions, verbs, adverbs contractions, determiners, modals etc.	(Alshamasi & Menai, 2022; Zlatkova et al., 2018; Zuo, Zhao & Banerjee, 2019)
Context Features	Key words, Interest groups, special activities	Abbasi & Chen, (2005)
Structural Features	linsear_write_formula, Flesch_kincaid_grade, Diversity, Dale_chale_readability , Automated_readability index	(Zlatkova et al., 2018; Zuo, Zhao & Banerjee, 2019)
	special character ratios, ratios of tabs, mean sentence length, average number of words, ratio of uppercase letters	(Sittar et al., 2016)
	average number of characters, average number of sentences beginning with uppercase, average number of sentences beginning with lower case	(Kaur et al., 2020)
	ratio of interrogative sentences	(Sittar et al., 2016)

From table 2.3, a total of 39 studies were reviewed for the stylometric features which have been used. The study defined five categories for the identified features as lexical, syntactic, character, structural and context features. All the features which were identified were grouped in these five categories. Studies which used feature combinations from different categories appeared in references of different feature categories to indicate that the said studies used those features. A total of 50 features were identified as tabulated in table 2.3.

2.5 Evaluation Metrics for Writing Style Change Detection

The fourth objective was to evaluate the effectiveness of ensembles of machine learning models in predicting the number of style changes in multi-authored documents at the sentence level. Numerous evaluation metrics exist in literature which can be used to verify the performance of the different methods for writing style change detection. The choice of an evaluation metric is dependent on the task at hand and the desired output. In literature measures such as accuracy score, F1score, Bcubed-F1 score and mean Average Precision have been used to evaluate the performance of the writing style change detection methods.

2.5.1 Accuracy

Accuracy is a measure of correctness of the algorithm's predictions of a writing style change detection method. It computes the number of correct predictions in relation to the total number of predictions. The focus was on the how well the algorithm was able to predict an instance relative to the total number of predictions made.

Accuracy can be broken down to its components by using the different prediction outcomes in a model. The possible scenarios in the binary classification performed in experiment one is; a positive observation predicted as positive known as True Positive (TP), a positive observation predicted as negative referred to as False Negative (FN), a negative observation predicted as negative known as True Negative (TN and a negative observation predicted as positive referred to as False Positive (FP).

Therefore, for binary classification tasks, accuracy can be calculated in terms of positives and Negatives as follows;

$$Accuracy = \frac{TP + TN}{TP+TN+FP+FN} \tag{2.1}$$

This measure is widely used in writing style change detection to separate single authored from multi-authored documents, although few studies have also been used to determine the number of style changes in a multi-authored document Akiva & Koppel (2012). Accuracy is used to measure the purity of writing style change detection algorithms such that higher accuracy values indicate that the model predicted most values correctly see (2.1). However, it cannot be adequately used alone in cases where the data set is not balanced (Nath, 2019; Rosso et al., 2016).

2.5.2 F1 Score

F1score is defined as the harmonic mean of precision and recall. The harmonic mean is an alternative metric for the more common arithmetic mean. It was computed as follows;

$$Y = 2 * \frac{P * R}{P + R} \qquad (2.2)$$

Where Y is the F1 score, P is Precision and R is the Recall

This measure is used when computing the average performance rate where there is more than one task. For instance, in Kocher, (2016) it was used to combine the precision of the algorithm on grouping documents written by the same author together, and the recall of grouping sections of a document written by the same author together. The overall performance of the method was determined by calculating the F1score, which is the average performance rate. While a higher F1score is desirable, a medium F1 value may require scrutiny to identify the type of errors.

Both precision and recall have the same weight in F1 measure. A high F1score is achievable if both recall and precision are high, while a low F1 value indicates that both recall and precision values are low. A medium F1 value is obtainable if either precision is high or recall is low and vice versa.

2.5.3 BCubed-F1 Measure

Bcubed-F1 scoring is based on performance evaluation of clusters. The precision and recall for each entity are calculated and then combined to produce the final precision and recall for the entire output (Kuznetsov et al., 2016; Safin & Kuznetsova, 2017).

For an entity i, the precision and recall are defined as follows;

$$r_i = \frac{c_i}{t_i} \qquad (2.3)$$

Where r_i is the recall for an entity i, C_i is the number of correct elements in the output chain containing I, and t_i is the number of elements in the true chain containing I.

Precision on the other hand is computed as follows;

$$p_i = \frac{c_i}{x_i} \qquad (2.4)$$

Where *C_i is the number of correct elements in the output chain containing i, and x_i is the number of elements in the output chain containing entity i.*

The final and precision and recall for all the entities are given by:

$$\mathrm{r} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{w_i} . \mathrm{r_i} \qquad (2.5)$$

While final precision for all the entities is given by”

$$\mathrm{p} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{w_i} . \mathrm{p_i} \qquad (2.6)$$

Where N is the number of entities in the document, and w_i is the weight assigned to entity i in the document.

Bcubed-F1 score is used to overcome the shortcomings of F1score where both recall and precision have the same weight, and therefore considers all types of errors to be equal.

2.5.4 Ordinal Classification Index (OCI)

Ordinal classification is a form of multi-class classification for which there is an inherent order between the classes, but not a meaningful numeric difference between them. The OCI measure used to measure the error of predicting the number of style changes for documents with multiple authors. Since it is a measure of the error rate, it is computed by calculating the Mean Absolute Error (MAE) which addresses the problem of ordinal classification as a regression problem (Nath, 2019; Zangerle et al., 2019; Zuo, Zhao & Banerjee, 2019). Cardosso and Sousa, (2011) computes OCI by calculating the Mean Absolute Error as shown in 2.8.

$$MAE = \frac{1}{N}\sum_{x\epsilon\sigma} \quad \left|g(e_x) - g\left(e^{\frown}{}_x\right)\right| \tag{2.8}$$

Where g(.) corresponds to the number assigned to a class, N = card and ex and $\widehat{e_x}$ *are the true and estimated values, x is the index vector.*

From (2.8), the OCI value is the inverted value of the MAE. The smaller the OCI value the better the performance.

2.6 Research Gap

The following gaps were identified and addressed;

Few studies had been done on detecting writing style changes where the styles change occurs between sentences within a paragraph. Most studies focused on detecting writing styles changes where the changes occur between paragraphs (Castro-Castro et al., 2020; Zuo, Zhao & Banerjee, 2019).

Few studies have used ensembles of machine learning algorithms in detecting writing style. ensembles of machine learning models (Nath, 2019; Zuo, Zhao & Banerjee, 2019).

Few studies are able to detect writing style changes in documents written by many authors (more than 3) where each author writes very short texts, in form of sentences which are randomly distributed in the document (Lyer & Vosoughi, 2020).

Few studies had been done to determine the optimal document and sentence level feature sets for detecting writing style changes in documents (Zlatkova et al., 2018; Sari, 2018).

2.7 Conceptual Framework

The conceptual framework of the study showing the relationship of variables is presented in figure 2.1.

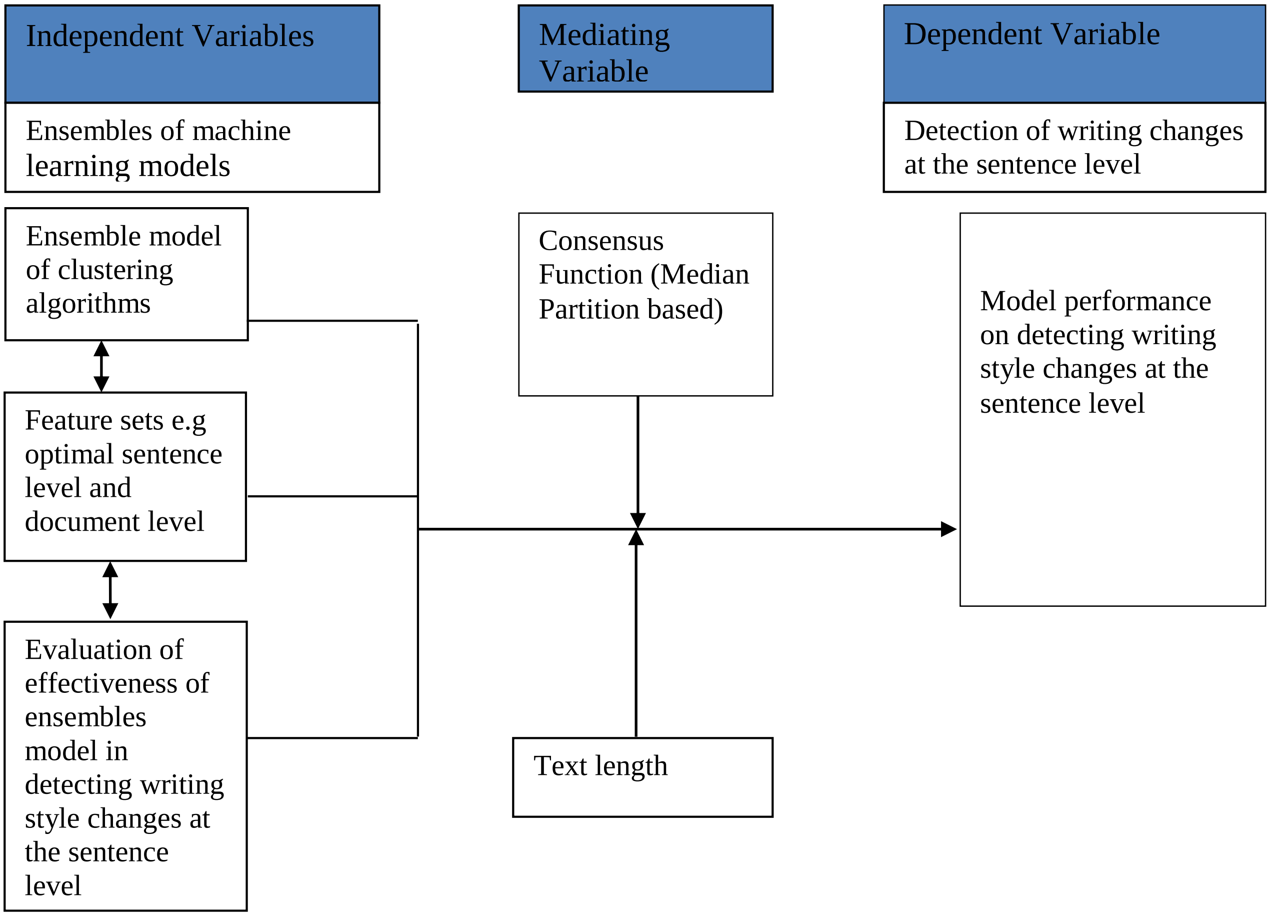

Independent Variables
Ensembles of machine learning models
Mediating Variable
Dependent Variable
Detection of writing changes at the sentence level
Ensemble model of clustering algorithms
Feature sets e.g optimal sentence level and document level
Evaluation of effectiveness of ensembles model in detecting writing style changes at the sentence level
Consensus Function (Median Partition based)
Text length
Model performance on detecting writing style changes at the sentence level

Figure 2.1: Conceptual Framework Showing relationship between the Ensembles models and Performance on Writing Style Change Detection

From figure 2.1, the independent variable was ensembles of machine learning models while the dependent variable was detection of writing style changes at the sentence level. Here, the independent variable directly affects the dependent variable. Specifically, the ensembles of machine learning algorithms models had a direct impact the detection of writing style changes at the sentence level. Similarly, the feature set used also affects the performance on short text such as a sentence. The mediating variables; the consensus function and the text length improve the performance on short text when models and feature sets are kept constant. The text length affects the strength of the relationship between the independent and the dependent variable. For instance, increasing the text length could improve performance on short text while reducing the text length could lead to a degradation of performance on short text.

2.8 Summary

Stylometry defines a rich set of stylometric features. Existing studies indicate that over 1000 stylometric features exist and continues to be used in authorship verification. These features fall under lexical, syntactic, character, structural and context features (Anwar et al., 2019; Castro-Castro et al., 2020; Gómez-Adorno et al., 2018; Potha & Stamatatos, 2018). However, it is argued that there exists no single feature capable of discriminating between works of different authors. The use of ensemble methods is believed to yield better results (Brocardo et al., 2015; Iqbal et al., 2010). Better still there is no optimized feature set combination. Therefore, feature selection is a vital step in authorship verification, and selecting appropriate features can help improve the of machine learning models on writing style change detection. Standardization of feature sets to specific problems may greatly improve research in these areas.

Multi-author analysis models are regularly employed to cluster together works of the same author, and form separate clusters for works of other authors (Amigud et al., 2017; Brocardo et al., 2015). This has mostly been studied as a multi-class text categorization problem that assumes that there is only one author with several documents. These studies continue to yield promising results on long documents where sufficient data exists for algorithm training and testing.

However, they ignore certain fundamental issues in authorship studies such as the fact that documents can be multi-authored and therefore each author could have written only very short pieces of text. Multi-author analysis is particularly challenging because it requires the verification of short texts, from a large set of authors. However, existing models suffer from similarity overlap as author numbers increase and the text length reduces.

Authorship verification on short texts has also been investigated. These studies indicate that authorship verification on short text is challenging. More particularly, uni-variate analysis of short text is even more challenging (Brocardo et al., 2013). However, this study believes that using numerous features may be fruitful in verifying authors of short texts. In addition, algorithms that can tackle similarity overlap is required.

On other hand, few studies have looked the multi-author analysis. These works have explored the possibility of identifying if a document is single authored or multi-authored. These studies have been done under the umbrella of style change detection, which strives to establish the number of authors of a document (Potthast et al, 2017). Different methods have been developed to solve this problem as illustrated in Akiva & Koppel, (2012), Castro-Castro et al., (n.d.) and Zhou & Wang, (n.d.). These studies used unsupervised learning techniques with lexical, syntactic and character features. The major limitation of all these studies is the fact that the number of authorship of a document is small and the text length is longer. Akiva & Koppel, (2012) argues that an increase in the number of writers in a multiple authored document leads to a decrease in classification accuracy.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

This chapter describes materials and methods used for the experimental development and evaluation of the ensemble models.

The chapter is organized as follows; Section 3.2 presents the research design. Section 3.3 highlights the model design, Section 3.4 outlines the model implementation. Section 3.5 outlines the determination of optimal feature sets. Section 3.6 describes evaluation metrics, Section 3.7 describes the ethical considerations, while Section 3.8 provides the chapter summary.

3.2 Research Design

The study used Mixed research design methods. This method comprised exploratory and experimental research designs. Mixed methods have the advantage of integrating quantitative and qualitative methods in a way that allows for more perfect and collaborative utilization of data as opposed to using either of the methods. Sequential mixed methods research study was used where qualitative data informs the quantitative experiments. This research design defines three stages of analysis: after the qualitative phase, after the secondary quantitative phase and at the integration phase that connects the two strands of data and extends the initial qualitative exploratory findings (Creswell & Clark 2011).

In the first phase of this Mixed Method Research study, the qualitative data analysis phase, a survey of the state-of-the-art studies for stylometric features applicable to writing style change detection was carried out. Specifically, stylometric features that have been used in authorship verification on short text and writing style change detection were considered. A survey of the stylometric features can be found in Oloo et al., (2022b).

Experimental research design was then used for the rest of the study such as feature engineering, separating single authored documents from multi-authored documents and in determining the number of style changes in multi-authored documents. The study used exploratory research to tackle part of objective 3 while experimental research design was used to tackle objective 1, objectives 2, part of objective 3 and objective.

3.3 Models Design

The number of algorithms used in the design of the two models was guided by the consensus function used and the availability of computing resources required. Since majority voting and median-based consensus function were used to combine results of the ensemble, a number was chosen such that there would be a clear win in terms of the number of votes received to declare a winner. Consequently, even numbers were not considered since they could result into a deadlock. Only odd numbers were considered such as three, five, seven, and so on. Because using a bigger number would result into more computing resource requirements, having more than 3 algorithms would require more computing resources and more training time thereby affecting the study which was constrained by time and resources. The study limited the number of base algorithms to three in each case.

The base algorithms of the ensemble were chosen based on their strengths in performing the task at hand drawn from literature, ability to complement each other and their computing resource requirements. For the two models ensemble of three algorithms was used.

3.3.1 Model Design for Separating Single Authored Documents from Multi-authored Documents

An ensemble model of three supervised learning algorithms was designed.Availability of labeled data and the fact that the study needed to accurately classify as many documents as possible necessitated the use of supervised learning. The ensemble's base algorithms were selected based on their capacity for binary classification, and representative in enhancing the performance of the ensemble as a whole. Three supervised learning algorithms were used; Support Vector Machines, Random Forest and Logistic Regression.

Support Vector Machines (SVM), for example, can analyze sparse and high dimensional data. SVM is capable of handling multi-class problems as well as binary classification problems by identifying the best separator (hyperplane) between two classes is incredibly resilient to outliers. The primary disadvantage of SVM is the is the long training time needed when large training data are available, however in this case the training data was not very large hence its suitability.

Random Forest classifiers perform better than the majority of base classifiers, including Decision Trees, SVM, Naïve Bayes, and Logistic Regression because its not affected by high dimensional data (Lyer & Vosoughi, 2020; Alvi, Algafri & Alqahtani, 2022). It was suitable because it uses different subsets of data to train the model hence eliminating biases in results.

Logistic regression is one of the most often used algorithms for binary classification to perform binary classification (Weerasinghe & Greenstadt, 2020; Pinzhakova, Yagel & Rabinovits, 2021). It presents its outputs in the form of 0s and 1s, which can be simply interpreted to mean the probability of each case belonging to a class. It was preferred in this study because of its strength in performing binary classification. This work maximized the advantages of each algorithm individually for increased accuracy and decreased variation by combining these three algorithms into an ensemble. The abstract architecture of the model design is shown in fig 3.1.

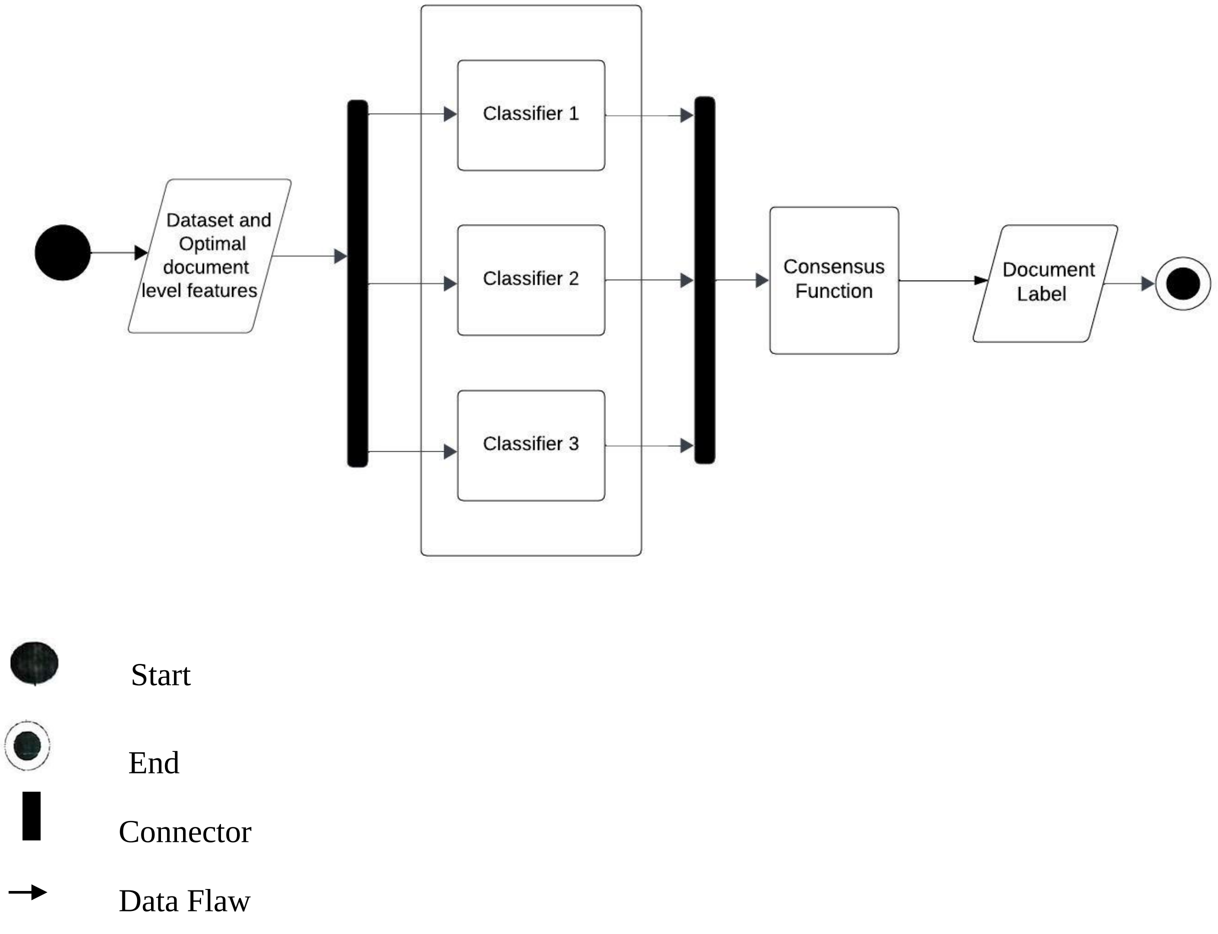


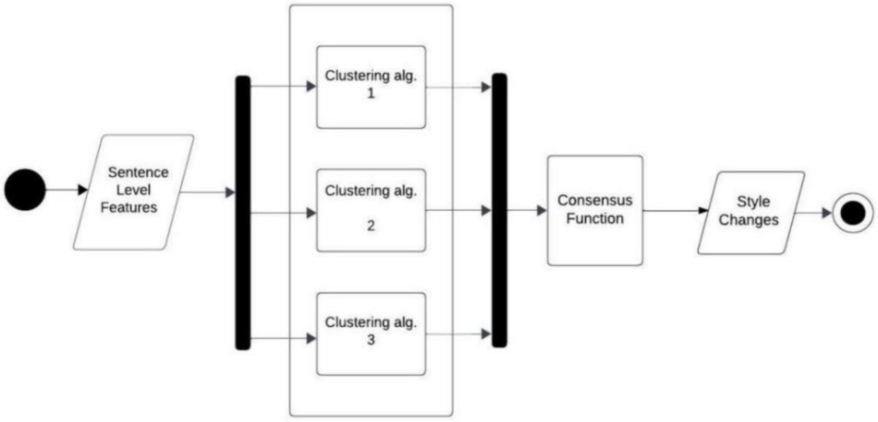
Figure 3.1: Model Design for Separating Single Authored Documents from Multi-Authored Documents

From figure 3.1, the input were the optimal document level features and all the documents in the data set, while the output was the document label indicating whether a document is single authored or multi-authored.

An ensemble of three classification algorithms was trained and used to determine whether a document had more than one writing style or not. Each classifier learned from the data based on the determined optimal document level features to give independent outputs. A consensus function based on majority voting was used to give the final prediction; indicating whether a document was single or multi-authored. The majority voting was used to pick the class label which was the majority or highest voted by the three classifiers. Documents which had more than one style were classified as label 1, while the ones which had only one writing style were classified as 0.

3.3.2 Model Design for Determining the number of Authors in Multi-authored Documents

The second task was designed to determine the number of authors in documents classified as multi-authored documents. The architecture of this model is shown in figure 3.2.



● Start

◉ End

▌ Connector

→ Data Flaw

Figure 3.2: The Model Design for Determining the Number of Writing Style Changes in Multi-Author Documents.

An ensemble model of three unsupervised learning algorithms; K-means, BIRCH and Gaussian Mixture Models was used in the design. Unsupervised learning was used so that the model could be applied in solving real-life problems where labeled data may be limited or lacking. The three algorithms were chosen based on the fact that they could perform hard clustering and the need to have different architectural representation in the model. Moreover, computing resource requirements was also considered such that algorithms which could use limited memory and processing power requirements were considered.

Three clustering algorithms- K-means, BIRCH clustering and Gaussian Mixture algorithms was used. The individual outputs of each algorithm were combined together using the median partition-based consensus function. K-means which is a widely used clustering algorithm due to its simplicity and good performance was used. It can perform both hard and soft clustering. In this study it was appropriate because of its good performance and ability to perform hard clustering. BIRCH clustering was used in this study because of its ability to form distinct clusters and its efficiency in terms of resource usage and runtime requirements. It is a partitioned Hierarchical clustering which is efficient in terms of resource usage and runtime requirements (Zhang, Ramakrishnan & Livny, 1997). Gaussian Mixture Algorithms were considered because it is a probabilistic algorithm which can be used for hard partition clustering with good performance. It was suitable in this study because of its difference in architectural design to two other algorithms hence ensuring representational of the results.

Here the outputs of the base algorithms were combined together using median-based consensus function. Members of the ensemble operate at the same level, each receiving the inputs to generate independent cluster labels for determining the number of authors. The cluster labels are fed into the consensus function, which then selects the definitive consensus one with the median mutual information score. The output of the consensus, which is the median cluster label, is used to determine the number of style changes.

3.4 Implementation of Ensemble of Machine Learning Models for Detecting Writing Style Change Detection

The second objective was to implement ensembles of machine learning models for detecting writing style changes in document. To realize this objective, two models were implemented. The first model was used to determine the optimal feature sets, and to separate single authored and multi-authored documents while the second model was used to determine the number of writing style changes in documents classified as multi-authored. Given that no single machine learning algorithm is superior to the others, ensembles which combines two or more algorithms can improve results of writing style change detection.

The models were implemented using Python version 3.9. A 64-bit windows 11 Pro operating system was used. The hardware requirements used included an x64-based processor. An Intel $^{(R)}$ core $^{(TM)}$ i7-8550U was used. The processor speed was between 1.9 GHz to 2.11GHZ. Memory included a Random-Access Memory of 8.00GB and a hard disk of 500GB.

3.4.1 data sets

This study used secondary data from a publicly available data set. The Pan @CLEF 2019(Zangerle et al., 2019) style change data set was used for experiments and evaluations. The data set requested from ZENODO was mined from the Stack Exchange network which contains various user posts in the form of question and answer. ZENODO is a multi-disciplinary open repository which is maintained by CERN. Pan data sets are commonly used public data sets for the style change detection tasks since they possess different characteristics simulating different environments and can be applied to several writing style change detection problems. The suitability of this data set was anchored on the availability of benchmark algorithms against which the proposed approach was validated. Moreover, the guarantee of data set quality and the provision of ground truth data was very advantageous to this study.

The data set was received in two batches; the first batch was a folder containing the training and validation sets. The validation set contained 25% of the whole data set while the training set contained 50% of the entire data set. Both the training and validation set consisted of ground truth information regarding the number of authors and locations of authorship switches. The training and validation sets were used for ensemble models training and validations.

A third set, the test set was received later after the ensemble models implementation and it was used to test and evaluate the ensembles of machine learning algorithms. The test set contained 25% of the whole data set however, it did not have the gold-standard labels. The training corpus consisted of 2544 documents, and a separate validation set containing 1272 documents. All the documents were written in English across different topics. For each document in the training and validation sets, gold-standard labels indicating author numbers and the annotations marking who authored exactly which portions of the document was provided.

Train-Test Split

This study used as train-test split of 75:25 split indicating that 75% of the data set was used for training and validation while 25% was used for testing. The train-test ratio was guided by the fact that the data set was smaller and there were so many hyper parameters to tune. Consequently, using larger ratios would have led to over-fitting and bias. In addition, the 75:25 ratio has been used effectively by previous studies.

The data set was split into a separate validation set to prevent the ensembles algorithms from over-fitting. This is a scenario where machine learning models become really good at classification and clustering the samples in the training set but cannot generalize and make accurate predictions on the data it has not seen before (Zangerle et al., 2019). Specifically, 50% of the data set was used for training, 25% of the data set was used for validation while 25% was used for testing.

Sample Size and Sampling Technique

The corpus comprised of various user posts in form of question and answer, mined from Stack-Exchange network. The posts covered different topics across different domains. The data set contained 5088 documents including single-authored and multi-authored documents. Half of the documents were single authored while the half was multi-authored. The documents in the data set were short length documents comprising short paragraphs. This data set has been used successfully by previous works on the task of writing style change detection (Nath, 2019; Zuo, Zhao & Banerjee, 2019).

This study used census study technique which involves using every item of the study population. Census is most preferred to sampling because its findings are more reliable and accurate especially with smaller populations. Deriving a representative sample is difficult when the population is too small or diverse in such cases census study is the only alternative (Mugenda & Mugenda, 2003). All the data points in the data set were used for either training, validating and testing the models. Whereas this data set was smaller than other data sets used in similar studies in terms of the number of documents, it was just sufficient for this study given that the study assumed a case where writing style changes happen at the sentence level thereby using sentence level features to determine the number of authors (Kestemont et al., 2018: 2019). In addition, most studies that used this data set applied the census technique yielding promising results (Nath, 2019; Zuo, Zhao & Banerjee, 2019).

Experimental Setups

Four experiments were setup. Experiment one was used to rank features to determine the optimal document level features. An ensemble model of three classifiers was used to rank features based on feature importance scores. The selected features, optimal sentence level features were used in experiment three to separate single authored from multi-authored documents.

Experiment two used the training data to rank sentence level features to determine the optimal sentence level features. An ensemble model of classifiers with the training data was used in this experiment. Optimal sentence level features were used to determine the number of authors in multi-authored documents.

Experiment three (3), was used to separate single authored documents from multi-authored documents by determining the existence of different writing styles in a document. Documents which exhibited a uniform writing style throughout were categorized as single authored while those that had more than one writing style were grouped as multi-authored documents. The ensemble model of clustering with the optimal document level features was used to determine the existence of style changes in documents. All the documents which were classified as single authored were dropped at this level. Only documents which were classified as multi-authored were used in the next experiment.

Experiment four (4) used the optimal sentence level features to determine the number of authors in multi-authored documents. In this experiment, only documents which were categorized as having style changes are used. The focus was to determine the number of style changes which translates to the number of authors in a document. For instance, documents which yielded two clusters are considered to have been written by two authors while the ones which yielded three clusters are deemed to have been written by three authors and so on. The study employed the ensemble model of clustering to detect the number of style changes in multi-authored documents.

3.5 Determining Optimal Feature Sets for Writing Style Change Detection

The third objective was to determine the optimal feature sets usable by the ensemble models for detecting writing style changes in documents. This objective was realized by performing experimental analysis on the features of table 2.3 (see section 2.4.3) to identify significant features based on feature importance scores. The training data set was used with all the identified features and an ensemble model of three classifiers to rank features based on their feature importance scores. Feature importance scores range from -1 to 1, with score greater than zero (> 0) considered significant to model performance while the scores of zero and below are insignificant.

Two experiments were carried out; Experiment one to determine the optimal document level features, and Experiment two to determine the optimal sentence level features. For experiment one, ensemble model of three classifiers was used to rank all the identified fifty (50) features (see table 2.3 section 2.4.3) and determine each feature's importance score. Majority voting was then used to select the optimal document level features. Features which yielded importance scores greater than zero in more than one algorithm of the ensemble were picked to the the optimal document level features. In this experiment, both document and sentence level features are used. In experiment two, only sentence level features were ranked and the optimal sentence level features were selected based on the fact that the features yielded importance scores above zero in more than one algorithm of the ensemble. i.e majority voting was used to select significant features considering that the feature yields importance score above zero in more than one algorithm.

3.6 Evaluating the Effectiveness of Ensemble Models in Detecting Writing Style Changes at the Sentence Level

The fourth objective was to evaluate the effectiveness of ensemble models in detecting writing style changes at the sentence level. Five evaluation metrics were used to assess the performance of the models; Rank, Precision, Recall, Accuracy, F1score and Ordinal Classification Index. Ranking was used to evaluate the model on the first and second experiment which sought to select the optimal document level and sentence level features. The third experiment which categorized single-authored documents from multi-authored documents was evaluated using Precision, Recall, Accuracy and F1score. Accuracy measure has been used as an evaluation tool in most classification studies (Apoorva & Sangeetha, 2021; Kumar et al., 2019; Sari, 2018; Brocardo et al., 2015). The possible scenarios in the binary classification performed; a positive observation predicted as positive known as True Positive (TP), a positive observation predicted as negative referred to as False Negative (FN), a negative observation predicted as negative known as True Negative (TN) and a negative observation predicted as positive referred to as False Positive (FP).

Therefore, accuracy was defined as the ratio of correct predictions for the evaluation data. It was calculated by dividing the number of correct predictions by the number of total predictions given by equation 2.2 (see section 2.5.2).

The value of the accuracy was between 0 and 1, where 1 indicates 100% that all the documents are correctly classified as either single author or multi-author, while 0 indicated the failure of the model to correctly classify even a single document. It is important to note that an accuracy of 1 is considered a theoretical value that may not be achieved by most algorithms.

Precision was determined by the following formula see (2.3) (section 2.5.3);

Precision was used to measure how precise or accurate a model is. In other words, out of the predicted positives how many of them are actual positives see (2.3) (section 2.5.3); While Recall was calculated as (2.4) (see section 2.5.3).

Recall captures the number of actual positives predicted as positive by an algorithm.

F1score is defined as the harmonic mean of precision and recall. The harmonic mean is an alternative metric for the more common arithmetic mean.

Both precision and recall have the same weight in F1 measure. A high F1score is achievable if both recall and precision are high, while a low F1 value indicates that both recall and precision values are low. A medium F1 value is obtainable if either precision is high or recall is low and vice versa. The fourth experiment was evaluated using the OCI measure. Ordinal classification is a form of multi-class classification for which there is an inherent order between the classes, but not a meaningful numeric difference between them. The OCI measure was used to measure the error of predicting the number of style changes for documents with multiple authors (Cardoso & Sousa, 2011; Zangerle et al., 2019). The study used Mean Absolute Error (MAE) which addresses the problem of ordinal classification as a regression task. The performance of the model was assessed in the data set as shown in (2.8) see section (2.5.4). The mean performance which captures the arithmetic mean of the accuracy and the inverted Ordinal Classification Index given by (3.1), was used to give the overall model performance and to compare with the performance of related studies;

$$mean = \frac{1}{n}\left(accuracy + (1 - OCI)\right) \qquad (3.1)$$

$$n = 2n = 2$$

The study used since only two performance vectors were used.

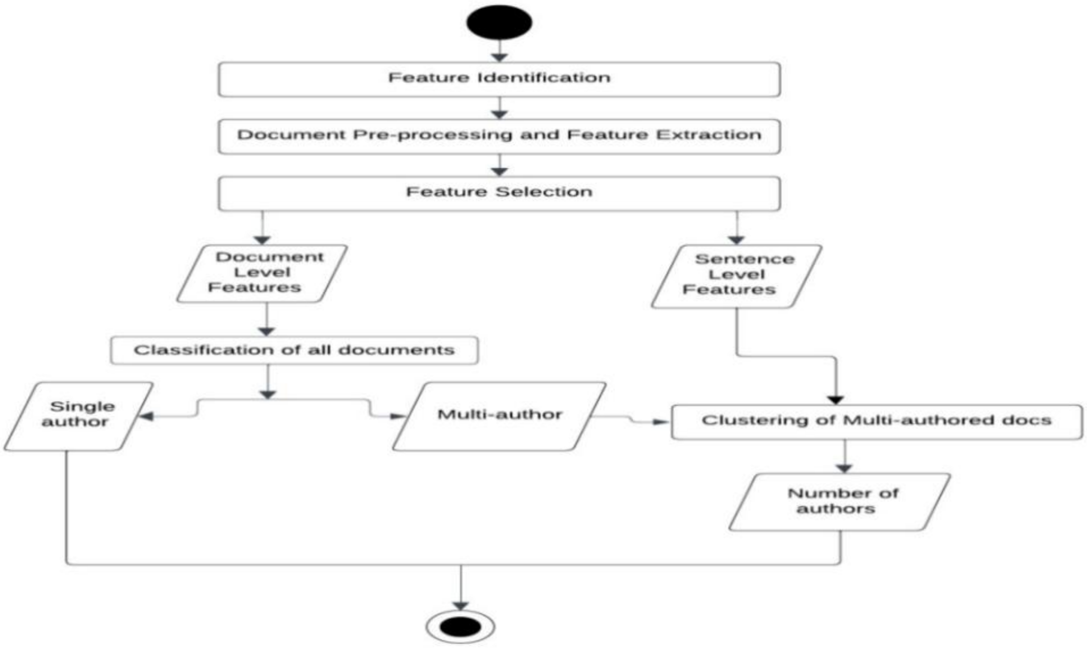
3.7 Ethical Considerations

This study considered the following specific ethical aspects:

Before commencement of the study, approval was sought by the researcher from Maseno University Ethics and Research Committee (MUERC) and National Commission for Science, Technology and Innovation (NACOSTI).

3.8 Chapter Summary

This chapter can be summarized diagrammatically by an activity diagram of fig 3.3.



● Start
◉ End
▌ Connector
→ Data Flaw

Figure 3.3: An Activity Diagram Summarizing Methodology

From figure 3.3, the study commenced by first identifying stylometric features which had been applied in writing style change detection and other related studies. Then the usual document processing was done involving feature extraction. At feature extraction, all the previously identified features were extracted; both the document and sentence level features were extracted. The next step was the feature selection where the optimal document and sentence level were selected based on an experimental analysis of their feature importance scores. The optimal document level features were used together with an ensemble model of classifiers was used to classify documents as either single authored or multi-authored, while the selected optimal sentence level features were used to determine the number of style changes in multi-authored documents.

CHAPTER FOUR

MODEL DEVELOPMENT

4.1 Introduction

This chapter discusses the development of the models. Description of the tools used to develop the models and the general information flow is presented. The rest of the chapter is organized as follows; Section 4.2 Model design, Section 4.3 highlights the models implementation, Section 4.4 outlines the feature engineering, 4.5 outline the models evaluation, and Section 4.6 gives the summary.

4.2 Model Design

The first objective of the study was to design ensembles of machine learning models for writing. Two models were designed; a model for separating single authored documents from multi-authored documents and a model for determining the number of authors in multi-authored documents.

4.2.1 Model Design for Separating Single Authored Documents from Multi-Authored Documents

This section presents the design of the model which was used in experiment three; to separate single authored documents from multi-authored documents. This experiment was conducted so as to reduce the amount of data which would be used with the clustering model. Since memory space was a challenge, doing several runs with the entire data set which included single authored documents would be time consuming. Therefore, to save on the time and memory requirements, the first tax involved separating documents so that only documents classified as multi-author would be used in the last experiment. The logical design for separating single authored from multi-authored documents is shown in figure 4.1.

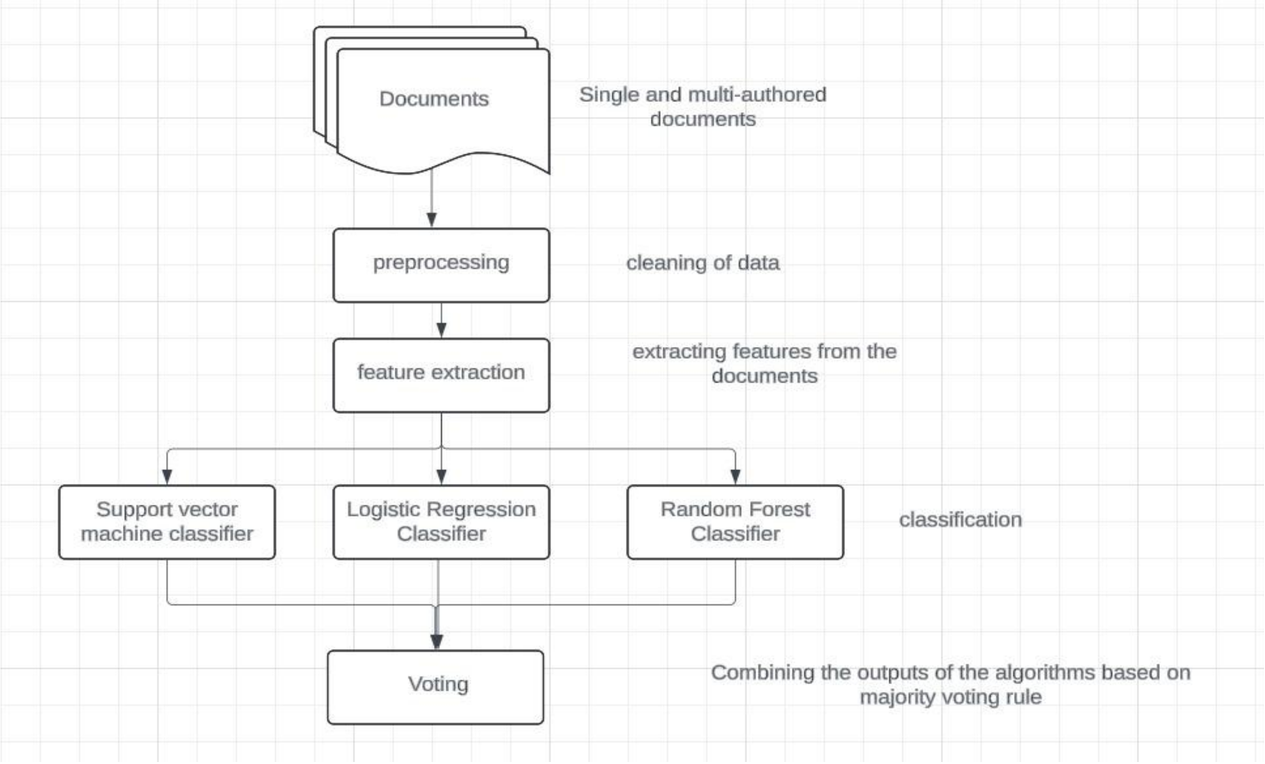

Documents
Single and multi-authored documents
preprocessing
cleaning of data
feature extraction
extracting features from the documents
Support vector machine classifier
Logistic Regression Classifier
Random Forest Classifier
classification
Voting
Combining the outputs of the algorithms based on majority voting rule

Figure 4.1: Model Design for Separating Single from Multi-Authored Documents.

Figure 4.1 represents the design of the model used to separate single authored from multi-authored documents.

4.2.1.1 Logistic Regression

A machine learning approach called logistic regression works well for classification tasks where the expected results are binary and the independent variables have little to no multicollinearity. Because the results were discrete, this study employed logistic regression as one of its ensemble members to distinguish between single- and multi-authored papers. It is based on calculating the likelihood that an event will occur or not; the likelihoods vary from 1 to 0. The study used the logistic function, a straightforward S-shaped curve that transforms data into a number between 0 and 1, to calculate the likelihood. The algorithm formulation was given by the function in (4.1):

The algorithm formulation was given by the function in

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \qquad (4.1)$$

Where $\beta_0 = -\frac{\mu}{s}$and is the intercept

$\beta_1 = \frac{1}{s}$ Is the inverse scale parameter or rate parameter, *s* is the scale parameter.

These are the y-intercept and slope of the log-odds as a function of x.

Conversely $\mu = -\frac{\beta_0}{\beta_1}$ and $s = \frac{1}{\beta_1}$

According to (4.1), all documents which had style changes were classified as having a value of 1 while the ones with no style changes were classified as having a value of zero. The interpretation of the algorithm was that a document with a value of 1 was classified as multi-authored while those with values of zero (0) were classified as single authored.

4.2.1.2 Support Vector Machine

The support vector machines used two parallel hyper-planes that split the data into two groups and kept as much space between them as feasible to classify documents as single or multi-authored. The maximum-margin hyperplane is the hyperplane that is halfway between the two hyper-planes, and the region delimited by them is referred to as the margin. These hyper-planes can be described by the following functions:

$$K(X_1, X_2) = (a + X_1^T X_2)^b \qquad (4.2)$$

where b is the degree of the kernel, a the constant term and x the set of points in the data set.

From equation (4.2), the document label was obtained by the label of the class in which it fell. There were only two classes; data points lying above the maximum-margin hyperplane, and the ones lying below the maximum-margin hyperplane. Documents with multiple styles were grouped in one class, and those without writing style changes were grouped in another class.

4.2.1.3 Random Forest Classifier

Random Forest classifier is a set of tree-structured classifiers, {*h*(*x*,*k*),*k* = 1,...}. The {\\k} are independently distributed random vectors, and each tree votes unitarily for the most popular class at input x. To complete the classification objective, it ensemble-assembles the outputs from several decision trees. Its strong performance and immunity to the dimensional curse made it a viable option for our investigation. In order to reduce the dimensional issues, the random forest learns by choosing a subset of the training data and characteristics for each decision tree in the forest.

As a Gini Index, the random forest was developed to determine how nodes on a Gini impurity is one minus the sum of the squared probabilities of each class as shown in:

$$Gini = 1 - \sum_{i=1}^{c} (pi)^2 \tag{4.3}$$

where C represent the number of classes *pi* represent the relative frequency of the class being observed in the data set.

The class and probability were used to determine the Gini of each branch on a node, determining which of the branches is more likely to occur.

4.2.1.4 The Consensus Functions

Results of the three algorithms were combined together using a consensus function based on majority voting as shown in fig 4.2. This study used hard majority voting where the predicted final class label is the class label that has been predicted most frequently by the classification algorithms. Hard voting is the simplest case of majority voting.

Hard voting is the simplest case of majority voting.

$$\hat{y} = mode\{C_1(x), C_2(x), \ldots, C_m(x)\} \tag{4.4}$$

Where C_1, C_2...C_m are the individual classifiers in the ensemble.

From (4.4), it was predicted the class label $\hat{y}$, via majority voting of each classifier C_m, such that

for each document, the label which is voted by majority (more than one vote) of the algorithms in the ensemble becomes the predicted document label.

4.2.2 Model Design for Detecting Writing Style Changes in Multi-authored Documents

The study designed an ensemble model of three clustering algorithms which was used to detect the number of style changes in multi-authored documents. The architecture of model is shown in figure 4.2.

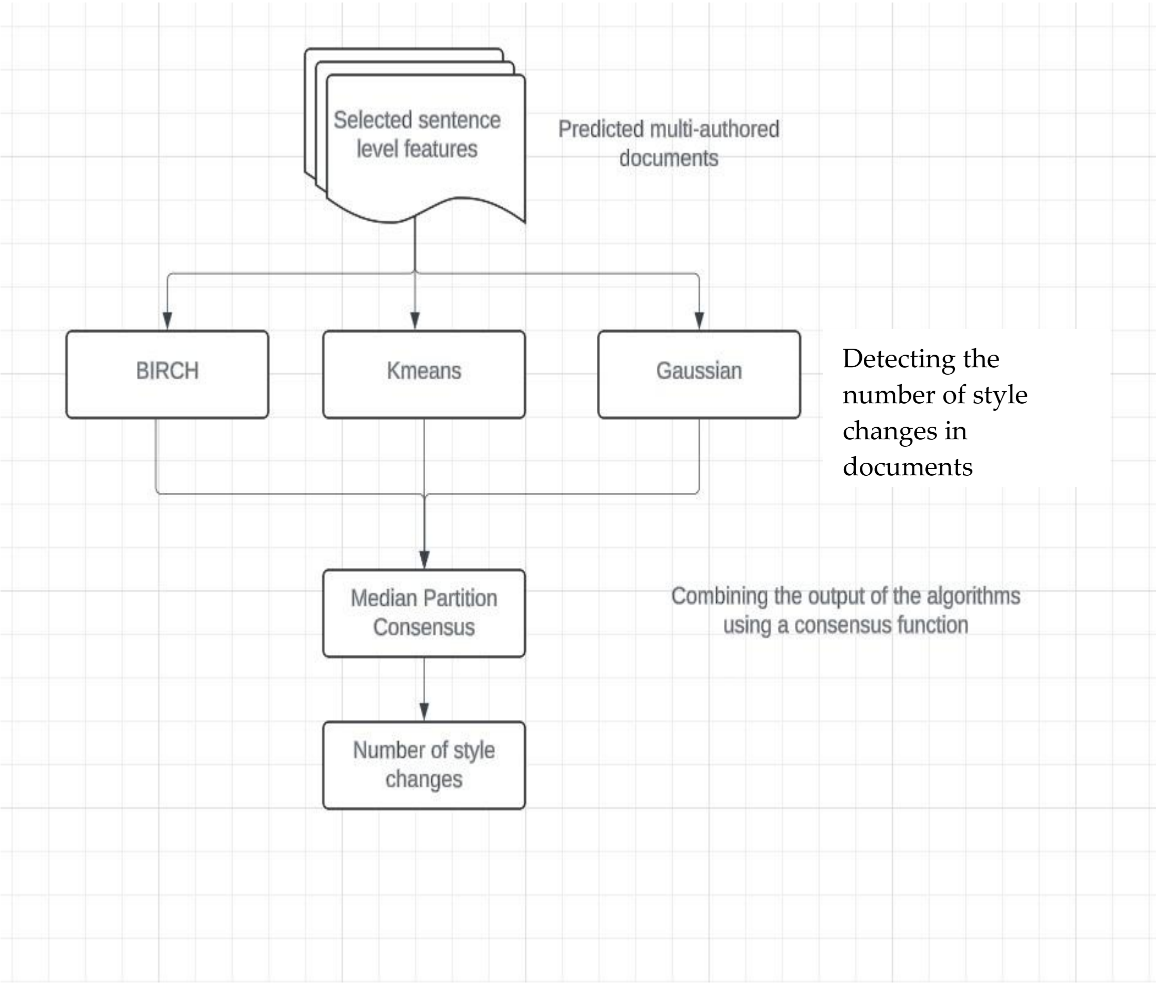

Selected sentence level features
Predicted multi-authored documents
BIRCH
Kmeans
Gaussian
Detecting the number of style changes in documents
Median Partition Consensus
Combining the output of the algorithms using a consensus function
Number of style changes

Figure 4.2: The Model Architecture for Detecting the Style changes in Documents.

Figure 4.2 outlines the architecture of the ensemble model of clustering. The ensemble model of three clustering algorithms; K-means, BIRCH and Gaussian Mixture Models each of which learn from the data to give independent outputs in terms of number of cluster labels in each document. The members of the ensemble were combined together using a median-based consensus function. Members of the ensemble operate at the same level, each receiving the inputs to generate independent cluster labels for determining the number of style changes. These cluster labels are fed into the super consensus, which then selects the definitive consensus one with the median mutual information score. The output of the consensus, which is the median cluster label is used to predict the number of style changes.

The input to the architecture was the set of optimal sentence level features and only those documents classified as multi-author. The output was cluster labels indicating the number of writing style changes in documents. The individual algorithms were formulated as below.

4.2.2.1 K-means Clustering

K-means algorithm, which is the most popular clustering algorithm due to its simplicity and effectiveness, was used. It divides data into k centroids and places all the data items into the nearest centroid. The goal is to choose a centroid which minimizes the inertia, calculated by measuring the distance between the data point and its centroid squaring the distances and adding them into one cluster. The main challenge in this algorithm is defining the value of K, which range from 1 to infinity. The study experimented with different values of K and settled at K=5 which was the highest number of authors in multi-authored documents in the data set. The output was a vector of integers indicating the cluster labels for each document in the data set. This study used the following algorithm to develop the K-means algorithm. The algorithm was adopted from Zhao et al., (2021).

Algorithm *(1)*

```
Choose k data points as the initial centroids(cluster centers)
repeat
    for each data point $\mathbf{x}$ $\epsilon$ do
        compute the distance from $\mathbf{x}$ to each centroid;
        assign $\mathbf{x}$ to the closest centroid // a centroid represents a cluster
    End for
    re-compute the centroids using the current cluster memberships
until the stopping criterion is met
```

According to (1), the algorithm randomly picks k (centroids). The centroids were given the labels c1, c2, … ck, such that C which is a set of all centroids was given by C= c1, c2, … ck

Assign each data point to its nearest centre by calculating the Euclidean distance

Here, the distance between each data point from each centroid is calculated and the data point is assigned to the centroid which yields the lowest value. This is repeated for all the data points.

The actual centroid is identified by taking the average of all the points assigned to that cluster given by:

$$ci = \frac{1}{|Si|} \sum_{xi \in Si} xi \qquad (4.5)$$

where *si* is the set of all points assigned to the i^{th} cluster.

According to 4.5, the original point, shifts to the new position, the actual centroid for each of these groups.

4.2.2.2 Gaussian Mixture

Hard clustering was accomplished using the probabilistic clustering algorithm known as Gaussian Mixture, which was appropriate because it allows for clusters with varying sizes and correlation structures within them. All parameters were set to default, with the exception of the number of clusters and random state, which were set at 42 for reproducibility of the results. The algorithm was developed by calculating the probability that a given data point, x, belongs to a particular cluster or component, k. The Gaussian equation was derived from the PDF function determination as follows: For a Gaussian mixture algorithm with K components, the kth component has a mean of μk and a variance of Σ (Dempster et al., 1977).

According to The mixture component weights are defined as ϕ_k for component C_k with the constraint that $\sum_{i=1}^{k} \phi_i$= 1 so that the total probability distribution normalizes to 1 such that;

$$p(x) = \sum_{i=1}^{k} \emptyset_i N \left(\frac{x}{\mu_i, \sigma_i} \right) \qquad (4.6)$$

Where k : the number of Gaussian components, N: number of data points and x, the dimensionality of the data.

The GMMs parameters: Means (μ) was used to indicate the center locations of Gaussian components, covariance matrices (Σ) defines the shape and spread of each component, while weights (ϕ_k) was used to give probability of selecting each component.GMM operates on the principle that a complex, multi-modal distribution can be approximated by a combination of simpler Gaussian distributions, each representing a different cluster within the data. The essence of GMM lies in its ability to determine cluster characteristics such as mean, variance, and weight. The mean of each Gaussian component represents a central point, around which the data points are most densely clustered.

The variance, on the other hand, provides insight into the spread or dispersion of the data points around this mean. A smaller variance indicates that the data points are closely clustered around the mean, while a larger variance suggests a more spread-out cluster. Expectation-Maximization (EM) technique was employed, alternating between the Expectation (E - step) and Maximization (M - step) steps until convergence. During the E - step, the model calculated the probability of each data point belonging to each Gaussian component. At the M - step, the model's parameters were adjusted based on the calculated probabilities at the E-step. GMMs cluster data points were based on the highest posterior probability.

4.2.2.3 BIRCH Clustering Algorithm

A type of hierarchical clustering technique called Balanced Iterative Reducing and Clustering utilizing Hierarchies (BIRCH) has been used in the past to handle noisy data. A set of N data points encoded as real-valued vectors and the desired number of clusters, K, are fed into the algorithm. It functions in four stages:

Using the data points, the first stage creates a height-balanced tree data structure known as a clustering feature (CF) tree. Given a set of N-dimensional data points, the clustering feature CF of the set is defined as the triple

$$\overrightarrow{LS} = \sum_{i=1}^{N} \overrightarrow{X_i} \qquad\qquad CF = (N, \overrightarrow{LS}, SS) \qquad\qquad (4.7)$$

Where is the linear sum is the square sum of data points.

$$SS = \sum_{i=1}^{N} (\overrightarrow{Xi})^2$$

Clustering features are organized in a CF tree, a height-balanced tree with two parameters: branching factor *B* and Threshold *T*. Each non-leaf node contains at most *B* entries of the form [CF_i, $child_i$] where $child_i$ is a pointer to its ith child node and CFi the clustering feature representing the associated sub cluster. A leaf node contains at most L entries each of the form [CFi]. It also has two pointers prev and next which are used to chain all leaf nodes together. The tree size depends on the parameter T. A node is required to fit in a page of size P. B and L are determined by P. so P can be varied for performance tuning. It is a very compact representation of the data set because each entry in a leaf node is not a single data point but a sub cluster.

In the second step the algorithm scans all the leaf entries in the initial CF tree to rebuild a smaller CF tree while removing outliers and grouping crowded sub clusters into larger ones. This study skipped this step because the data set wasn't very large and therefore all the data points were considered important.

Step three involves clustering every leaf entry using an established clustering technique. The sub clusters in this case are directly subjected to an agglomeration hierarchical clustering algorithm on the basis of their CF vectors. It gave the user the option to choose between the desired diameter threshold for clusters and the desired number of clusters. Following this stage, a collection of clusters representing the data's main distribution patterns is produced. Nevertheless, there may be a few small, localized errors that can be fixed with an optional step 4.

To create a new set of clusters, step 4 uses the centroids of the clusters created in step 3 as seeds, redistributing the data points to the nearest seeds. The option to exclude outliers is also provided in this phase. A point that deviates excessively from its nearest seed may be deemed an anomaly.

4.2.2.4 The Consensus Function

The consensus function was used to determine the partition P* of data set X by combining the members of the ensemble {P_1, P_2, P_3,…, P_m}, with a consensus function F without going back to the original features, so that P* is better than either P_1, P_2, … P_m.

The consensus function was used to combine the outputs of the three clustering algorithms to obtain the cluster partition P*, the final model result. In determining the final partition P*, the study adopted the median partition approach, which picks the partition with the highest similarity with all the partitions in the ensemble (Vega & Shulcloper, 2011).

The median partition was determined such that the sum of the dissimilarities between this partition and all the partitions determined by the ensemble members based on the selected features is minimal. The median partition P* was determined based on the description and the formula given below.

Let q be a set of partitions and ϖ be a similarity measure such as distance between two partitions. The study adopted the formula by Xu and Tian, (2015) to compute the median partition as shown in Equation (15).

$$P^* = argmin\ P\ \sum_{i=1}^{q} \omega(P_i, P) \tag{4.8}$$

This formula was used to determine the final model partition output, which represents the model prediction to establish the number of writers in documents. Equation (15) was applied to each partition to determine whether it similar to the others on not.

According to 4.8, the model prediction was represented by the partition that is closest to all the other partitions of the ensemble members. The median partition solves the problem of similarity overlap.

4.3 Implementation of Ensemble Models for Determining Writing Style Changes

To realize this objective two models were implemented. The ensemble model of classifiers and ensemble model of clustering. The ensemble model of classifiers was used to separate single authored documents from multi-authored documents while the ensemble model of clustering was used determine the number of authors in multi-authored documents.

4.3.1Model Construction for Separating Single Authored from Multi-authored Documents

The ensemble model of classifiers was constructed by setting parameters of individual algorithms to ensure performance optimization. For performance optimization the following hyper parameters were tuned and set at a level that provided best performance. The support vector machines was set to class output mode, the class was set to 8 (c:8), gamma: scale, kernel: poly and tol: 0.1. The kernel function was set to polynomial. Polynomial kernel was used because it works best in models where the data has nonlinear patterns or there are interactions between features. In this study, a number of features were used to determine similarities in writing styles of authors both at the sentence level and at the document level. Therefore to achieve best performance, a function that uses features and their combinations was the most suited, in this case polynomial kernel function.

The Logistic regression was based on the probability output. The class c:1, penalty: 11, and solver: lib-linear. Random forest on the other hand was used a probability output. The most important hyper parameters in random forest that were tuned were the number of trees in the forest (n_estimators) and the number of features considered for splitting at each leaf-node (max_features).

Optimum performance was achieved when the two hyper parameters were set at n_estimators= 100 and max_features = ‘sqrt’. Other parameters such as the maximum number of levels in each decision tree (max-depth was set at 8), the function to measure the quality of a split (criterion) was retained at the default “gini”. Random state (random_state) was set to an integer value, 2025 to control the randomness of bootstrapping of the samples used when building trees. Minimum number of data points placed in a node before the node is split (min_samples_split) was set at 5, minimum number of data points allowed in a leaf node (min_samples_leaf) was set at 2.

Experiment III: Separating Single - from Multi-authored Documents

The focus of this experiment was to separate single authored- from multi-authored documents. An ensemble of three classifiers consisting of SVM, Logistic Regression and Random Forest was used. The input was the set of optimal document level features and all the documents in the data set. The output was the document label indicating whether a document had style changes or not. Documents with style changes were labeled as 1 while a label of zero was for documents without style changes. Consequently, documents with style changes were classified as multi-authored while documents without style changes were classified as single authored. Results of the three algorithms were combined together using a consensus function based on majority voting. This study used hard majority voting where the predicted final class label is the class label that has been predicted most frequently by the classification algorithms.The model formulation was based on the individual algorithms’ formulation.

4.3.2 Model Construction for Detecting Writing Style Changes in Multi-authored Documents

The model for detecting writing style changes in multi-authored documents was constructed by setting and tuning hyper-parameter of individual algorithms. K means was constructed as follows; it was implemented using the K-means ++ nitialization (init=k-means++). K-means++ initialization selects initial cluster centroid based on an empirical probability distribution to the overall inertia. It was best suited in this study because it speeds up convergence.

The number of clusters to form as well as the number of centroids to generate, n_cluster variable was set to (n-clusters=5).

The n_init variable defines the number of times the K-means algorithm is run with different centroid seeds. The final results is the best output of the consecutive runs in-terms of inertia. The n_init was set to 9. The K-means algorithm was run nine (5) times with different centroid seeds.

Random-state variable, which when set to an integer value makes the results of the K-means reproducible in all the runs. It determines random number generation for centroid initialization. The random state variable was set to an integer value to make the randomness deterministic. In this study, the random state variable was set at (random state=42) 42, which is among the popular integer random seed.

BIRCH was constructed by setting n_clusters=cluster, threshold=0.8-The radius of the sub cluster obtained by merging a new sample and the closest sub cluster should be lesser than the threshold. Otherwise a new sub cluster is started. Setting this value to be very low promotes splitting and vice-versa.

Branching parameter defines the maximum number of CF sub clusters in each node. If a new samples enters such that the number of sub clusters exceed the branching factor then that node is split into two nodes with the sub clusters redistributed in each. The parent sub cluster of that node is removed and two new sub clusters are added as parents of the 2 split nodes. The branching factor was set at 45,

while Gaussian Mixture’s parameters were set to n components = cluster. This is the number of mixture components, random state=42, covariance type =sperical. This is a string describing the type of covariance parameters to use. Spherical requires each component to have its own single variance, init params = kmeas++ and n-init=6, number of initializations to perform, the best results are kept.

Experiment IV: Determining the Number of Authors in Multi-authored Documents

To determine the number of authors participating in writing multi-authored documents, an ensemble model of clustering algorithms was designed and developed. The ensemble model of three clustering algorithms K-means, BIRCH and Gaussian Mixture Models each of which learn from the data to give independent outputs in terms of number of cluster labels in each document. The base algorithms of the ensemble were combined together using a median-based consensus function. All the base algorithms of the ensemble operate at the same level, each receiving the inputs to generate independent cluster labels for determining the number of style changes. These cluster labels are fed into the super consensus, which then selects the definitive consensus one with the median mutual information score. The output of the consensus, which is the median cluster label is used to predict the number of style changes. The input to the model was the set of optimal sentence level features and only those documents classified as multi-author. The output was cluster labels indicating the number of writing style changes in documents.

4.4 Determining the Optimal Feature Sets for Writing Style Change Detection

In objective 3, the study sought to determine the optimal feature sets for writing style change detection. The focus here was to determine the optimal sentence- and document-level features for usable by the ensembles of machine learning models for detecting writing style changes in documents. Stylometric features that have been utilized in previous research for the purpose of detecting changes in writing style were used (see table 2.3 section 2.4.3).To select optimal feature sets for every task, text processing was carried out first, then feature extraction and selection.

4.4.1 Document Pre-processing

The study did not perform extensive document pre-processing on the data set (Brocardo et al., 2015). URLs and other technical details, such "OSX 10.11.2," were eliminated, nevertheless. Contracted word forms—distinctive word types that combine shortened versions of two or more words that typically go together, like "I have,""can't," and "not"—were employed as a feature in this study, they were first expanded before stop words were removed to enable their later extraction. The subsequent fundamental document processing was carried out including tokenization, lowercasing, removal of stop words and lemmatization.

I. Tokenization

Documents were tokenized first at the word and character level, so as to generate numerical features. Python's Natural Language Toolkit (NLTK) was used to tokenize the sentences. NLTK is a powerful suite that contains libraries and programs for statistical language processing (Ouyang et al., 2020; Sari, 2018).

II. Lower-casing

All upper-case letters were changed to lowercase as part of feature dimensionality reduction.

III. Removal of stop words

Words such as the, this, a, an, in, etc are regarded as stop words in the English language because they appear most frequently in all texts written in the language. Despite this, it is argued that they carry little lexical content and therefore may not be effective in differentiating different styles. However, this study investigated the use of contracted word forms which forms part of stop words library as a feature in establishing writing style changes within documents. Stop words were removed from the document using Natural Language Tool Kit library.

IV. Lemmatization

Lemmatization is the process of reducing a word to its base root. For purposes of dimensionality reduction, NLTK's Word Net Lemmatizer was used to reduce each word to its base root.

4.4.2 Features Extraction

Features belonging to the categories of lexical, syntactic, structural, character, and content aspects were extracted at the sentence level and document level. Sentence level features were used to determine the precise number of style changes in a document, whilst document level features were extracted to determine whether a document had style changes or not. The following feature extraction methods were used; TF-IDF Vectorization, Part-of Speech Tagging and various text statistics. These methods were used because of they could handle the size of the data set used in the study and also because of their efficiency in extraction important features.

TF-IDF vectorizer was used to extract lexical features. This feature extraction method was used because of its ability to provide importance of words in addition to providing frequency of words in a corpus. Word, phrase, and character-level lexical features were extracted using NLTK's TF-IDF vectorizer. The proportions of different types of lexical items in the document were employed as features in the study. In particular, each lexical feature’s total number of occurrences was tallied and divided by the total number of components in the document. Eleven word-based features, seven sentence-level features, and thirteen character-based features were among the features. Overall, 18 lexical features at the word, phrase, and sentence level were used. On the other hand 13 character levels were used. (see table 1 section 3.1).

Part-Of-Speech tagging was also used in the study to extract Part-Of-Speech words. This technique was used to determine a sentence’s syntactic structure and identify each words role in a sentence. Specifically, the rule-based Part-Of-Speech tagging was used used because of its simplicity in terms of implementation and, because it does not require a large amount of training data. The POS tagger from NLTK was used to extract POS terms. Function terms such nouns, pronouns, prepositions, adjectives, interjections, conjunctions, verbs, adverbs, contractions, determiners, and modals were extracted using the POS tagger. The TF-IDF vectorizer was used to extract punctuation. The punctuation and special symbols such as punctuation based on Unicode format, including single quotes, commas, periods, colons, semi-colons, question marks, exclamation points, and special marks were extracted. Nineteen (19) syntactic elements in all were taken out and utilized in the study. (see table 2.3 section 2.4.3).

Various text statistics were used to extract structural features. The primary structural features extracted for this study were readability measures, along with other data like mean sentence length and average word length. The readability features Linsear-write formula, given by; Linsear-write formula was used to gauge the readability of documents based on sentence length and number of words used that have three or more syllables. The formula used was as follows;

$$\mathrm{LWF} = \frac{[(100-(\frac{100*p}{nw}))+(3*\frac{100*q}{nw})]}{100*\frac{nst}{nw}}$$

Where p= number of words with less than 3 syllables,

q= number of words with 3 syllables or more, nw is number of words in a sentence and

nst is number of sentences.

Flesch-Kincaid grade is measure that was used to judge the readability level of various documents it was calculated using the following formula

$$\text{Flesch} - \text{Kincaid grade} = 0.39\left(\frac{\text{words}}{\text{sentences}}\right) + 11.8\left(\frac{\text{syllables}}{\text{words}}\right) - 15.59 \qquad (4.9)$$

Automated Readability Index (ARI) was computed based on the formula provided below.

$$\text{ARI} = 4.71\left(\frac{\text{characters}}{\text{words}}\right) + 0.5\left(\frac{\text{words}}{\text{sentences}}\right) - 21.43 \qquad (4.10)$$

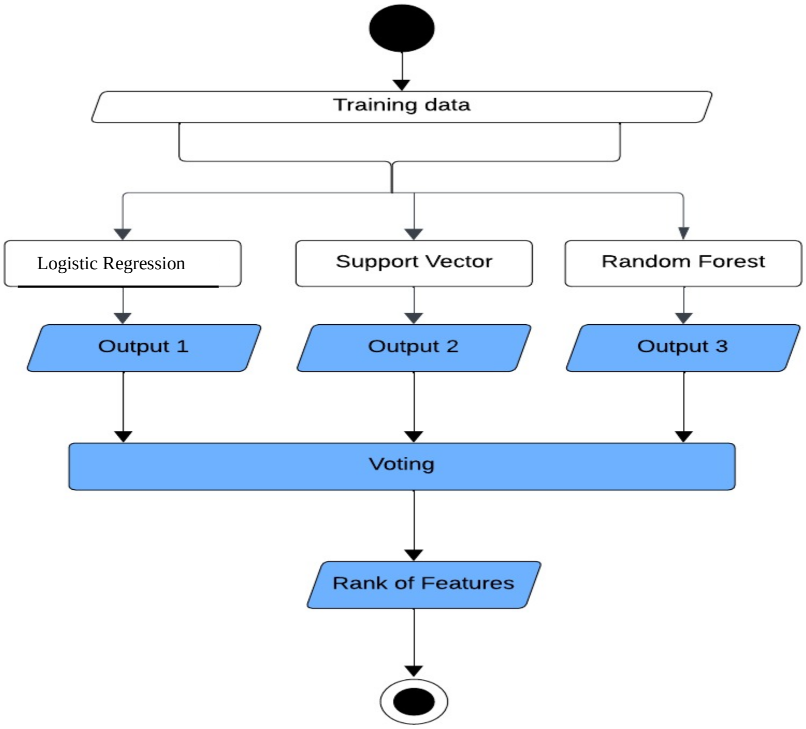
Automated Readability Index was used to gauge the understandability of text.

Dale-Chall-Readability, SMOG grade, Coleman-Liau index, tough words, and Gunning-Fog were computed and extracted in this study using the Text stat Pythons package. The rest of the features such as tab ratios, special character ratios, the ratio of uppercase letters, the average word count, the average sentence length, the average character count, the average number of sentences starting with uppercase, the average number of sentences starting with lowercase, and the ratio of interrogative sentences were extracted by collecting the total number of occurrences and dividing by the entire number of words in the document, (see table 2.3 section 2.4.3).

4.4.3 Feature Selection

Feature selection was carried out using experimental methods. This method is believed to be better than manual methods of feature selection because it determines the actual importance of a feature for the task at hand. This study performed two experiments to select the optimal feature sets to be used in the rest of the study. Experiment (1) to select the optimal document level features and Experiment II to select the optimal sentence level features. The ensemble model of classifiers was used together with the features of table 2.3 (see section 2.4.3)

The architecture of the model used to rank features comprised of three classifiers; Support Vector Machines, Random Forest and Logistic Regression whose outputs were combined together using majority voting. Availability of labeled data and the fact that the study needed to the best features as possible necessitated the use of supervised learning. The ensemble's base algorithms were selected based on their capacity for binary classification, and representational in enhancing the performance of the ensemble as a whole. The structure of the model used to rank features based on feature importance scores is shown in figure 4.3.



● Start
◉ End
▮ Connector
→ Data Flaw

Figure 4.3: Model Design for Ranking Optimal Document and Sentence Level Features

From figure 4.3, the input consisted of all the identified features from literature and all the documents in the training set. The output was a list showing all the features and their ranks.

Three classifiers were used whose outputs were combined together using majority voting. Majority voting was used to select the optimal document and sentence level features.

4.4.4 Experimental Setups

Two (2) experiments were carried out. Experiment I was to identify document level features able to determine the existence of writing style changes in documents. The output of this experiment was used to separate single authored from multi-authored. Experiment II to identify optimal sentence level features and to select the optimal feature set for determining the number of style changes in multi-authored documents. Experiments I and II were done autonomously, experiment II following Experiment I with different sets of inputs.

I. Experiment I: Determining the optimal document level features

Finding the most important document-level features that could be utilized to detect style changes in documents was the focus of this experiment. In order to achieve this, the study used their computed feature importance scores to rank all of the extracted features at the document and sentence levels. The features that yielded importance scores greater than zero (0) were considered important to the model performance. An ensemble model of classifiers was trained on all the features discovered in the literature, as described in table 2.3 (see section 2.4.3), in order to compute feature significance scores and rank features. Feature importance is used to assign each input feature of a given algorithm a score, with the scores simply indicating the weight of each feature. A higher score indicates that the particular feature will have more of an impact on the model being used to forecast a particular variable. The drop-column feature importance strategy was used to determine the feature importance scores based on the following algorithm:

```
Algorithm feature Importance(C, F_train, F_test):                    (2)
Input: C classifier // the classifier used
        F_train N*M matrix of N features for M documents // the training data
        F_test N*M matrix of N features for M documents  // the testing data
Output: list S of feature importance scores //

model <- C.train(F_train)
baseline_score<-model. test(F_test) // importance score when all the features are used
```

```
for f¬ 0 to N -1 do
     i.   F_train_drop_1 <-del  F_train[f]
    ii.   F_test_drop_1 <-del  F_test[f]
   iii.   model <- C.train (F_ F_train_drop_1)
    iv.   score_drop_1<-model. test(F_test_drop_1) // importance score minus the dropped
          feature
     v.   S <-difference(baseline_score, score_drop_1) // overall feature importance score
    vi.   return S
```

This algorithm was used to highlight how drop-column feature importance score method which was used in the two experiments works. In this strategy, baseline performance score is computed first by using all the features and sample of the training data. Model performance is noted down, this is the baseline performance. Next a feature is dropped, the model is trained and performance is measured. This is performance of the model without that feature. To obtain the performance of each feature, the difference between the baseline performance and the model performance minus the feature is computed. This value gives the importance score for that feature. The process iterates for all the features. This method was applied for both sentence level features and document level features.

According to (2), the model is first trained on all the features and performance is captured. Then a feature or column is removed entirely from the data set, the model is retrained, and the impact on performance is assessed. Essentially, the plan is to obtain a baseline performance score using permutation significance, remove a feature completely, retrain the model, and then recalculate the score. The difference between the baseline and the model's performance when a feature is absent determines the feature's importance score. This technique provided an even more direct answer than the permutation importance strategy to the question of how significant a feature is to the overall performance of the model. Despite its high processing cost, prior research has effectively extracted significant features using this approach, leading previous studies to see it as the most effective (Koppel & Schler, 2004; Ghosh et al., 2016). In this experiment, an ensemble model consisting of three classifiers based on majority voting was employed. The model design is shown in fig 4.1 (see fig 4.1 section 4.2.1).

In this experiment, all the machine learning algorithms operate at the same level, each receiving the set of features and generating independent feature importance scores as the output. Majority voting consensus function was then used to generate the final feature importance scores and to rank features. The output of this experiment was a table indicating features and their importance scores ranked in order of increasing importance.

II. Experiment II: Optimal Sentence Level Feature Identification Experiment

This experiment sought to determine the optimal sentence level features which would then be used to determine the number of writing style changes in documents. The ensemble of three classifiers was used in this experiment as detailed in fig 4.1(see section 4.2.1).

To realize this, the study ranked all the extracted features both at sentence level based on their feature importance scores. The features which yielded feature importance scores above zero were considered the most significant. An ensemble of three supervised algorithms was trained on all the sentence level features identified in literature and extracted as listed in table 2.3 (see section 2.4.3). This experiment was based on algorithm Feature Importance (C, F_train, F_Test) (see experiment I). This algorithm was used to determine the feature importance score in almost a similar manner as in the first experiment except that the model was trained sentence level. Therefore at the beginning the model was trained on all the sentence level features and the model performance captured. Then a feature was entirely removed from the data set, the model retrained and the model performance captured. To determine the actual importance of the feature, the difference between the baseline performance, when all features are used, and the performance of the model without that feature was computed. The value obtained translated to the importance of that feature. Essentially, the plan is to obtain a baseline performance score using permutation significance, remove a feature completely, retrain the model, and then recalculate the score. This was done for all the features.

4.5 Model Evaluation

The test data set was used to evaluate the performance of the models in each experiment independently. The corpus consisted of 1272 documents with no gold-standard labels indicating the number of authors and where authorship switches occurs in each document. Before testing, all the documents were split into corresponding sentences. Padding with zeros was used where the sentence length was too short to standardize all sentence lengths. Five evaluation metrics were used; Ranking, Accuracy, F1score, Precision, Recall and an OCI measure. Ranking was used to evaluate experiment I and II, to select the optimal document and sentence level features. Accuracy, Precision, Recall, and F1score were used to evaluate experiment III, while Ordinal Classification Index measure was used in experiment IV.

4.5.1 Model Bench-marking

Bench-marking of the models were done in relation to the models developed by Zuo, Zhao & Banerjee, (2019) and Nath, (2019) on writing style change detection. These were the top performing models on determining the number of authors in multi-authored documents during the 2019 CLEF. They were based on the Pan @Clef 2019 style change data set, which was the data set used in this study. The study defined a baseline accuracy of 0.5 (random guess) and a more informed baseline accuracy of 0.512 for experiment III i.e separating documents single-authored documents from multi-authored documents. For experiment IV, the study defined the random guess baseline OCI of 1.0 and a more informed baseline OCI of 0.849. The more informed baseline performance was determined using the most frequent feature for each task. The most frequent feature used in experiment one was question sentences. While the most frequent feature for experiment two was total words.

The model was bench-marked with two top performing models on the PAN @CLEF 2019 style change detection data set. The suitability of these models lied on the fact that they obtained accuracy of 0.6 and 0.649 respectively which were above the baseline accuracy of 0.5 (random guess) for the first experiment. They also achieved top performance in the second experiment obtaining OCI measures of 0.808 and 0.847 respective which were slightly above the baseline OCI of 0.849 (more informed baseline). This study compared the results of the proposed models with the results of the benchmark algorithms on the same data set.

4.6 Summary

The study's main emphasis in this chapter was on creating and putting into practice models for multi-author analyses. To achieve this, two tasks were completed: identifying the number of writing style changes in publications classed as multi-authored and categorizing documents as single or multi-authored. The classification challenge involved identifying the number of style changes in multi-authored documents, whereas the clustering effort involved differentiating single-authored materials from multi-authored documents. In order to accomplish these goals, this study first designed features by determining the optimal feature set for each task through experimental investigation of existing features in the literature.

Feature engineering was carried out to select the optimal feature set for the classification task and the clustering task. It was based on identified features from literature which had been applied in authorship verification involving short text and writing style change detection tasks. The identified features were ranked based on their feature importance scores using an ensemble of three (3) supervised classifiers selected based on their simplicity and effectiveness in classification tasks. Specifically, Support Vector Machines, Logistic Regression and Random Forest which have been shown to perform comparatively well in classification tasks were used. The results of the three algorithms were combined together using majority voting rule to give the final results of the algorithm or the ensemble. In this algorithm each individual algorithm learned on the training data set to give an independent list of ranked features based on the drop column feature importance. The final feature ranks were obtained through majority voting where a feature was considered important if it was important in more than one algorithm.

For the multi-author analysis, two algorithms were designed and implemented. The implementation of the algorithm was based on python programming language. The first algorithm; an ensemble of three supervised machine learning algorithm namely Support Vector machines, Logistic Regression and Random Forest was used as in feature engineering. Their outputs were combined together using majority voting. Here the optimal document level features were used with the ensemble to identify the existence of different writing style in documents. The documents which exhibited signs of more than one writing style were classified as multi-authored and those which did not as single authored. The main focus here was to correctly classify as many documents as possible hence the use of supervised learning algorithms. For this task, an accuracy measure was used to evaluate the performance of the algorithm on the test data set.

The clustering task involved determining the number of style changes in documents classified as multi-authored from the previous experiments. The optimal sentence level features were used with an ensemble of three clustering algorithms to group together sentences with similar writing styles. The document was first broken down into its constituent sentences. It was assumed that each sentence is written by just one author. The algorithm was used to group together similar sentences into one cluster. Each cluster contained documents written by one author, with the total number of clusters indicating the number of style changes in a document. Because of the short text length used, the median partition consensus function was used to combine the results of individual algorithm together to give the final algorithm prediction. This consensus function was based on determining the distances between the cluster labels. The partition with the smallest distance between it with the others was picked as the final algorithm partition and was used for prediction. For this task, an OCI measure was used to evaluate the algorithm. The algorithm bench-marking was based on comparing results of existing related algorithms which were used to solve the same task on this data set. consequently, two algorithms were identified upon which the proposed approach was validated on. The two algorithms were the approaches by Nath, (2019) and Zuo, Zhao & Banerjee, (2019). The detailed comparison of the performance of the proposed algorithms and existing ones is provided in chapter 5.

CHAPTER FIVE

RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter presents the results of the experiments carried out. A discussion of the results is also done and the findings highlighted. The rest of the chapter is organized as follows; section 5.2 presents the results of feature selection, section 5.3 discusses the results Model evaluation.

5.2 Feature Selection

The third objective was to determine optimal feature set for detecting writing style changes in documents. The study determined the optimal feature set at both the document and sentence level, and presents the results as follows.

5.2.1 Optimal Document Level Features

By rating each feature and choosing the ones with the greatest importance scores, the study attempted to determine the best document-level features. Despite using an ensemble of three supervised learning algorithms, the study first shows the outcomes of each method separately before presenting the ensemble's findings. The normalized feature significance scores range from negative one (-1) to one (1). According to this method, a feature's importance to the performance of the model increases with its feature importance score. In particular, features are deemed significant when their importance score is larger than zero, and less significant when it is less than or equal to zero. Negative feature importance scores (importance near 0) imply that the feature does not significantly influence predictions. In this regard, features whose importance (scores > 0) were considered significant while those with scores less than or equal to zero were not significant. The results of feature ranking using random forest are shown in table 5.1.

Optimal Document Level Features Using Random Forest

All the identified features from literature were ranked using Random Forest Classifier and the results presented in table 5.1.

Table 5.1: Random Forest Feature Ranking.

SNO	Features	Importance score	S_No	Feature	Importance Score
1	Digits	0.003142	26	Word_len_two_and_three	-0.001178
2	Question sentences	0.003142	27	Smog index	-0.001178
3	Coordinating conjunction	0.001964	28	Word_len_gte_six	-0.001178
4	Check uppercase	0.001571	29	Linsear write formula	-0.001571
5	diversity	0.001178	30	Personal pronouns	-0.001571
6	adjectives	0.001178	31	modals	-0.001571
7	Parenthesis count	0.001178	32	Mean word length	-0.001964
8	First word uppercase	0.001178	33	Short sentences	-0.001964
9	Punctuation count	0.001178	34	5_gram	-0.001964
10	Interjections	0.000786	35	trigram	-0.001964
11	Prepositions	0.000393	36	bigram	-0.001964
12	Check available vowel	0.000393	37	unigram	-0.002357
13	Alphabets	0.000000	38	Num_sent	-0.002357
14	contractions	0.000000	39	Difficult words	-0.002357
15	Special character	0.000000	40	Total words	-0.002357
16	Comma count	0.000000	41	adverbs	-0.002357
17	Flesch reading ease	-0.000393	42	Long sentences	-0.002357
18	Num_sentence_repetition	-0.000393	43	emoji	-0.002749
19	Mean sentence length	-0.000393	44	4_gram	-0.002749
20	pronouns	-0.000786	45	Sentence_upper_begin	-0.002749
21	Flesch Kincaid grade	-0.000786	46	Semicolon count	-0.002749
22	Determiners	-0.000786	47	Colon count	-0.003142
23	nouns	-0.001178	48	Sentence_begin_lower	-0.003142
24	verbs	-0.001178	49	Dale-Chall readability score	-0.003535
25	Type token ratio	-0.001178	50	Automated readability index	-0.003928

Table 5.1: Presents the feature importance scores and their ranks using Random Forest classier.

From table 5.1, it can be seen that twelve (12) features yielded feature importance scores of above zero. They include digits, question sentences, coordinating conjunctions, check uppercase, diversity, adjectives, parenthesis count, first word uppercase, punctuation count, interjections, prepositions, and check available vowels. These features were regarded as determinant for this algorithm's prediction. The rest of the features yielded feature importance scores of zero (0) and below. They included the use of alphabets, stop words such as contractions, number of special characters, frequency of using comma, readability measures and scores such as the flesch reading ease, Flesch Kincaid grade, SMOG index, Linsear write formula, Dale-Chall readability score and Automated readability. Others include the number of sentence repetitions, mean sentence length, pronouns, determiners, nouns, verbs, words with lengths of two and three, words with lengths of six, personal pronouns, modals, mean word length, short sentences, 5-grams, trigram, bigrams, unigrams, number of sentences, difficult words, total number of words, adverbs, long sentences, the use of emojis, 4-grams and number of sentences beginning with uppercase letters. These features were regarded as insignificant to the algorithm performance because they yielded feature importance scores of between 0 and -003928.

The highest ranked features in top three positions were digits and question_sentences which yielded the same importance score and coordinating conjunctions. The least significant features were Automated Readability score, Dale-Chall readability, number of sentences beginning with lowercase and the number of colons. All the four least significant features had negative feature importance scores.

In conclusion the optimal document level features according to Random Forest Classifier were: digits, question sentences, coordinating conjunctions, check uppercase, diversity, adjectives, parenthesis count, first word uppercase, punctuation count, interjections, prepositions, and check available vowels**.**

Support Vector Machines

Results of feature ranking using Support Vector Machines are presented in table 5.2.

Table 5.2: Document Level Feature Ranks using SVM

S_No	Features	Importance Score	S_No	Features	Importance Score
1	Colon_count	0.003142	26	Flesch_kincaid_grade	0.000000
2	5_gram	0.002357	27	Word_len_two_and_three	0.000000
3	Punctuation_count	0.001964	28	Sentence_begin_upper	0.000000
4	Total words	0.001964	29	Short sentences	0.000000
5	Personal pronouns	0.001178	30	nouns	0.000000
6	Check uppercase	0.001178	31	Sentence begin lower	0.000000
7	adverbs	0.001178	32	Mean word length	0.000000
8	prepositions	0.001178	33	Special character	0.000000
9	diversity	0.001178	34	Semicolon count	0.000000
10	First word uppercase	0.001178	35	contractions	0.000000
11	4_gram	0.001178	36	emoji	0.000000
12	verbs	0.000786	37	Check available vowel	0.000000
13	pronouns	0.000786	38	adjectives	0.000000
14	unigram	0.000393	39	Word_len_gte_six	0.000000
15	Parenthesis count	0.000393	40	Type token ratio	0.000000
16	trigram	0.000393	41	alphabet	0.000000
17	Coordinating conjunctions	0.000393	42	bigram	-0.000393
18	Linsear write formula	0.000393	43	Question sentences	-0.000393
19	Comma count	0.000393	44	interjections	-0.000393
20	Difficult words	0.000393	45	Flesch Reading ease	-0.000393
21	SMOG index	0.000393	46	Dale-Chall readability	-0.000393
22	digits	0.000393	47	determiners	-0.000786
23	Mean sentence length	0.000000	48	modals	-0.000786
24	Long sentences	0.000000	49	Num_sent	-0.001178
25	Automated Readability index	0.000000	50	Num_sentence_repetition	-0.002357

The results of table 5.2 show that twenty-two (22) features had feature importance scores of above zero. These features were: number of colons, 5_gram, number of punctuation used, total number of words, personal pronouns, the use of Upper-casing, adverbs, prepositions, diversity, number of first words uppercase, 4_gram, verbs, pronouns, uni-gram, parenthesis_count, tri-gram, coordinating_conjunctions, linsear_write_formula, comma_count, and difficult_words.

Features with importance scores of zero and below were use of alphabets, stop words such as contractions, number of special characters, readability measures and scores such as the Flesch reading ease, Flesch-Kincaid grade, SMOG index, Dale-Chall readability score and Automated readability. Others include the number of sentence repetitions, mean sentence length, pronouns, determiners, nouns, verbs, words with lengths of two and three, words with lengths of six, modals, mean word length, short sentences, bi-gram, uni-grams, number of sentences, long sentences, the use of emojis, number of sentences beginning with uppercase letters, sentence begin lower, semicolon count, check available vowels, adjectives, type token ratio, question sentences, interjections. The top three most significant features according to SVM were colon count, 5_grams and punctuation count. While the least significant features were; number of sentence repetitions, num_sent and modals. Therefore, the optimal document level features using Support Vector Machines were: number of colons, 5_gram, number of punctuation used, total number of words, personal_pronouns, the use of Upper-casing, adverbs, prepositions, diversity, number of first words uppercase, 4_gram, verbs, pronouns, uni-gram, parenthesis count, tri-gram, coordinating conjunctions, Linsear write formula, comma count, and difficult words.

I. **Logistic Regression**

The results of ranking document level features using Logistic Regression classifier is presented in table 5.3.

Table 5.3: Document Level Feature Ranks using Logistic Regression

S_No	Features	Importance Score	S_No	Features	Importance Score
1	Difficult words	0.007855	26	Mean sentence length	0.000000
2	Question sentences	0.005106	27	trigram	0.000000
3	5_gram	0.003142	28	Dale-Chall_readability_score	0.000000
4	4_gram	0.002749	29	SMOG index	0.000000
5	Total words	0.001964	30	num_sent	0.000000
6	Colon count	0.001964	31	num_sentence_repetition	0.000000
7	prepositions	0.001571	32	contractions	0.000000
8	Check available vowels	0.001178	33	determiners	0.000000
9	Personal pronouns	0.001178	34	Coordinating conjunctions	0.000000
10	alphabets	0.000786	35	pronouns	0.000000
11	adverbs	0.000786	36	Punctuation count	0.000000
121	digits	0.000786	37	Comma count	0.000000
13	Short sentences	0.000393	38	Semicolon count	0.000000
14	verbs	0.000393	39	emoji	0.000000
15	nouns	0.000393	40	interjections	-0.000393
16	modals	0.000393	41	Special character	-0.000393
17	Long sentences	0.000393	42	First word uppercase	-0.000393
18	adjectives	0.000393	43	Parenthesis count	-0.000393
19	Sentence begin lower	0.000393	44	Mean word length	-0.000393
20	Flesch Kincaid grade	0.000393	45	word_len_gte_six	-0.000786
21	diversity	0.000393	46	Uni-gram	-0.000786
22	word_len_two_and_three	0.000393	47	Linsear write formula	-0.000786
23	Sentence begin upper	0.000393	48	Type token ratio	-0.000786
24	Flesch_reading_ease	0.000000	49	bi-gram	-0.001178
25	Automated readability index	0.000000	50	Check uppercase	-0.001178

Results of table 5.3 shows that a total of twenty-three features had feature importance scores of above zero (0). These features include difficult words, question sentences, 5_grams, 4_grams, total words, colon count, prepositions, check available vowels, personal pronouns, alphabets, adverbs, digits, short sentences, verbs, nouns, modals, long sentences, adjectives, sentence begin lower, Flesch Kincaid grade, diversity, word_len_two_and_three and sentence begin upper. The rest of the features had features yielded zero (0) or negative importance scores. Features with zero (0) and negative importance scores were considered to have no significance to the performance of the algorithm and were therefore not important. Less significant features according to the results of table 5.3 were Flesch reading ease, mean sentence length, Automated readability score, tri-gram, SMOG Index, Dale-Chall readability score, num_sent, num sentence repetitions, contractions, determiners, coordinating conjunctions, pronouns, punctuation count, comma count, semicolon count, emoji, interjections, special characters, first word uppercase, parenthesis count, mean word length, word_len_gte_six, uni-gram, Linsear write formula, type token ratio, bi-gram, and check uppercase.

The most significant features in top three positions were difficult words, question sentences and 5_grams. While the least significant features were check uppercase, bi-grams and type token ratio.

II. Optimal Document Level Features Using Ensembles Model

The final model prediction of the most significant features was drawn from the results of the three algorithms by selecting features which had majority votes, in this case more than one vote. The results of majority voting are shown in table 5.4.

Table 5.4: Results of document level features based on majority voting

S_no	Best Document Level Features	Number of votes
1	Colon count	2(SVM, LR)
2	5_gram	2(SVM, LR)
3	Punctuation count	2(SVM, RF)
4	Total words	2(SVM, LR)
5	Personal pronouns	2(SVM, LR)
6	Check uppercase	2(SVM, RF)
7	Adverbs	2(SVM, LR)
8	Prepositions	3(consensus)
9	Diversity	3(consensus)
10	First word uppercase	2(SVM, RF)
11	4-grams	2(SVM, LR)
12	Verbs	2(SVM, LR)
13	Parenthesis count	2(SVM, RF)
14	Coordinating conjunctions	2(SVM, RF)
15	Difficult words	2(SVM, LR)
16	Question sentences	2(RF, LR)
17	Check available vowels	2(RF, LR)
18	Digits	3(Consensus)
19	Adjectives	2(RF, LR)

The study noted from the results presented in table 5.4, that all the three algorithms had consensus on only three features which had feature importance scores of above zero. These features were prepositions, diversity and digits. Similarly, it can be seen that a total of nineteen (19) document level features were ranked as important by majority voting, and were considered as the optimal document level feature set for the task of separating single authored documents from multi-authored documents.

Therefore, the optimal document level feature set consisted of colon count, 5_gram, punctuation count, total words, personal pronouns, check uppercase, adverbs, prepositions, diversity, first word uppercase, 4_grams, verbs, parenthesis count, coordinating conjunctions, difficult words, question sentences, check available vowels, digits and adjectives.

5.2.2 The Optimal Sentence Level Features

To identify the optimal sentence level features, all the sentence level features identified from literature were ranked based on their feature importance scores using the ensemble model of three supervised algorithms; random forest classifier, Support Vector Machines and Logistic Regression. Each algorithm learn’t from the data to give independent feature importance scores for each sentence level features, and consequently ranked all these features based on their feature importance scores. A total of 32 sentence level features were subjected to the experiment to obtain their importance scores and to rank them based on their importance scores. The study present results of the individual algorithms, followed by the final ensemble model results.

Support Vector Machines Sentence Level Features Ranks

The result of sentence level feature ranks and importance scores using Support Vector Machines is shown in table 5.5.

Table 5.5: Sentence Level Features Ranking using Support Vector Machines.

S_NO	Features	Importance
1	Total words	0.004690
2	Adverbs	0.003752
3	Word_len_gte_six	0.003752
4	Diversity	0.002814
5	Parenthesis count	0.001876
6	Word_len_two_and_three	0.001876
7	Digits	0.000938
8	Sentence begin upper	0.000938
9	Semicolon count	0.000938
10	Colon count	0.000938
11	Long sentences	0.000938
12	Alphabet	0.000938
13	Type token ratio	0.000938
14	Modals	0.000938
15	Coordinating conjunctions	0.000938
16	Question sentences	0.000000
17	Special character	0.000000
18	First word uppercase	0.000000
19	Check uppercase	0.000000
20	Interjections	-0.000938
21	Verbs	-0.000938
22	Mean_word_length	-0.000938
23	Check available vowel	-0.000938
24	Punctuation count	-0.001876
25	Prepositions	-0.001876
26	Nouns	-0.001876
27	Personal pronouns	-0.001876
28	Pronouns	-0.001876
29	Difficult words	-0.001876
30	Determiners	-0.002814
31	Comma count	-0.002814
32	Adjectives	-0.004690

From the results of table 5.5 a total of fifteen (15) sentence level features yielded importance scores above zero, thereby considered important to the algorithm prediction. These features include; total words, adverbs, word_len_gte_six, diversity, parenthesis count, word_len_two_and_three, digits, sentence begin upper, semicolon count, comma count, long sentences, alphabets, type token ratio, modals, and coordinating conjunction. The remaining seventeen (17) features had importance scores of zero and negative values indicating that they were not important features for algorithm prediction. They included question sentences, special character, first word uppercase, check uppercase, interjections, verbs, mean word length, check available vowel, punctuation count, prepositions, nouns, personal pronouns, pronouns, difficult words, determiners, comma count, and adjectives.

The most significant sentence level feature was total words, which was the highest ranked based on this algorithm while the least ranked feature was adjectives. This algorithm considered almost half of the sentence level features as important while the other half were considered insignificant to algorithm prediction. In terms of percentages, 46.9% of the features were considered important while 53.1% were insignificant.

Logistic Regression Sentence Level Features Ranks

The results of ranking sentence level features using Logistic Regression is presented in table 5.6.

Table 5.6: Sentence Level Feature Ranks based on Logistic Regression

S_NO	Features	Importance
1	Modals	0.007505
2	Adverbs	0.007505
3	Total words	0.007505
4	Type token ratio	0.006567
5	Word_len_gte_six	0.006567
6	Question sentences	0.005629
7	Digits	0.004690
8	Semicolon count	0.004690
9	Nouns	0.004690
10	Parenthesis count	0.003752
11	Personal pronouns	0.003752
12	Diversity	0.002814
13	Prepositions	0.002814
14	Sentence begin upper	0.002814
15	Long sentences	0.001876
16	Difficult words	0.001876
17	Coordinating conjunction	0.001876
18	Interjections	0.001876
19	Comma count	0.001876
20	Word_len_two_and_three	0.000938
21	Pronouns	0.000938
22	Verbs	0.000000

23	Special character	0.000000
24	Determiners	0.000000
25	First word uppercase	0.000000
26	Check uppercase	0.000000
27	Alphabet	0.000000
28	Check available vowel	0.000000
29	Colon count	0.000000
30	Punctuation count	-0.001876
31	Mean word length	-0.002814
32	Adjectives	-0.022514

From the results presented in table 5.6, twenty-one features yielded feature importance scores of above zero (0) thereby considered significant to algorithm performance. The features were; modals, adverbs, total words, type token ratio, word_len_gte_six, question sentences, digits, semicolon count, nouns, parenthesis count, personal pronouns, diversity, prepositions, sentence begin upper, long sentences, difficult words, coordinating conjunction, interjections, comma count, word_len_two_and_three, and pronouns. The rest of the features yielded importance scores of zero and below and were regarded as less significant. These features include verbs, special characters, determiners, first word uppercase, check uppercase, alphabet, check available vowel, colon count, punctuation count, mean word length, and adjectives In terms of percentages 65.6% of the features were significant while 34.4% were insignificant. The most significant features were modals, adverbs and total_words while the least significant features were punctuation count, mean word length and adjectives. Overall, the best sentence level features according to Logistic Regression were modals, adverbs, total words, type token ratio, word_len_gte_six, question sentences, digits, semicolon count, nouns, parenthesis count, personal pronouns, diversity, prepositions, sentence begin upper, long sentences, difficult words, coordinating conjunction, interjections, comma count, word_len_two_and_three, and pronouns.

Random Forest Sentence Level Feature ranks

All the thirty-two (32) sentence level features were ranked using Random Forest classifier and the results presented in table 5.7.

Table 5.7: Sentence Level feature ranks using Random forest

S_NO	Features	Importance
1	Mean word length	0.006567
2	Verbs	0.006567
3	Check available vowel	0.006567
4	Long sentences	0.006567
5	Word_len_two_and_three	0.005629
6	Semicolon count	0.004690

7	Sentence begin upper	0.004690
8	Difficult words	0.004690
9	Modals	0.004690
10	Total_words	0.004690
11	Digits	0.003752
12	Coordinating conjunctions	0.003752
13	Nouns	0.003752
14	Interjections	0.003752
15	Question sentences	0.003752
16	Pronouns	0.003752
17	Prepositions	0.003752
18	First word uppercase	0.003752
19	Colon count	0.003752
20	Adverbs	0.002814
21	Determiners	0.002814
22	Type token ratio	0.002814
23	Check_uppercase	0.002814
24	Personal pronouns	0.002814
25	Comma_count	0.002814
26	Word_len_gte_six	0.002814
27	Special character	0.001876
28	Punctuation count	0.001876
29	Diversity	0.001876
30	Alphabet	0.001876
31	Parenthesis_count	0.000000
32	Adjectives	-0.008443

The results of table 5.7 indicate the results of ranking features using random forest classifier. From the results, almost all the features were considered significant- yielding feature importance scores of above zero while only two (2) features were insignificant. Significant features were modals, adverbs, total words, type token ratio, word_len_gte_six, question sentences, digits, semicolon count, nouns, personal pronouns, diversity, prepositions, sentence begin upper, long sentences, difficult words, coordinating conjunction, interjections, comma count, word_len_two_and_three, pronouns, verbs, special characters, determiners, first word uppercase, check uppercase, alphabet, check available vowel, colon count, punctuation count, and mean word length. Only two features had importance scores of zero (0) and below thereby considered insignificant. The features were adjectives and parenthesis count.

In terms of percentages, important features were 93.8% while only 6.2% were ranked as insignificant. The most significant features were mean word length, verbs, check available vowel and long sentences. The least ranking features were parenthesis count and adjectives. Random forest had the highest percentage of features ranked as important compared to SVM and LR which had 46.9% and 65.6%.

Ensembles Model Optimal Sentence Level Features

The optimal sentence level features for this model were determined through majority voting of the individual algorithms' best sentence features. Features which were voted by more than one algorithm as significant were considered as the optimal feature set. Results of ensembles model achieved through majority voting is presented in table 5.8.

Table 5. 8: Results of majority voting on Sentence Level features.

S_No	Features	Votes
1	Total words	3
2	Adverbs	3
3	Word_len_gte_six	3
4	Diversity	3
5	Parenthesis count	2(SVM,LR)
6	Word_len_two_and_three	3
7	Digits	3
8	Sentence begin upper	3
9	Semicolon count	3
10	Colon count	2(SVM,RF)
11	Long sentences	3
12	Alphabets	2(SVM,RF)
13	Type_token_ratio	3
14	Modals	3
15	Coordinating conjunction	3
16	Question sentences	2(LR,RF)
17	Nouns	2(LR,RF)
18	Difficult words	2(LR,RF)
19	Interjections	2(LR,RF)
20	Comma count	2(LR,RF)
21	Pronouns	2(LR,RF)
22	Prepositions	2(LR,RF)

From table 5.8, a total of 22 sentence level features were selected as significant through majority voting. These features were total words, adverbs, word_len_gte_six, diversity, parenthesis count, word_len_two_and_three, digits, sentence begin upper, semicolon count, colon count, long sentences, alphabets, type token ratio, modals, coordinating conjunction, question sentences, nouns, prepositions, difficult words, interjections, comma count, prepositions and pronouns.

The rest of the features were considered less significant because they either were ranked important by only one algorithm thereby getting only one vote, or they were considered less significant by all the three algorithms. These features were mean word length, verbs, check available vowels, interjections, first word uppercase, determiners, check uppercase, personal pronouns, special characters, punctuation count and adjectives.

Overal, the optimal sentence level features were given by total words, adverbs, word_len_gte_six, diversity, parenthesis count, word_len_two_and_three, digits, sentence begin upper, semicolon count, colon count, long sentences, alphabets, type token ratio, modals, coordinating conjunction, question sentences, nouns, prepositions, difficult words, interjections, comma count, prepositions and pronouns.

5.2.3 Discussion

The study noted that some features which ranked top in one algorithm were ranked almost at the bottom of the list by other algorithms. For instance, difficult words which was the highest-ranking feature in Logistic Regression was insignificant according to Support Vector Machines and insignificant by Random Forest. Moreover, 5_grams which was among the top three features in both SVM and LR, ranked poorly in Random Forest appearing as least significant feature. The study also noted that all the top three ranked features in the three algorithms appeared in the final algorithm most determinant document level features, a factor that reinforced the suitability of ensemble learning algorithms in feature selection. Conversely, not all features deemed important by one algorithm were ultimately important based on the ensemble model.

From the results of table 5.4, a total of 19 features were selected through majority voting as the most determinant document level features. Only features which appeared in two or more list of individual algorithms' significant features were considered as the final best document level features of the ensemble. It can be seen that most features got two votes while only three features got three votes (consensus). SVM and LR agreed on 8 features namely colon count, 5_grams, total words, personal pronouns, adverbs, 4_grams and difficult words. The two algorithms realized an overlap of 42.1% in significant features. SVM and RF on the other hand agreed on 5 features as significant namely; punctuation count, check uppercase, first word uppercase, parenthesis count and coordinating conjunction.

This overlap translates to 26.3%. Finally, RF and LR had a consensus on only three features: question sentences, check available vowels and adjectives. This amounted to 15.8% overlap in significant features. The main contributing factor to this kind of results was that each individual algorithm has a different architecture upon which decisions were made. However, there are minor similarities in LR and SVM architectures when the task at hand involves structured binary classification thereby explaining the reasons in higher overlap in significant document level features as opposed to RF and LR or RF and SVM. In addition, RF algorithm does not perform well when the number of decision points is two, like in this case deciding whether the document is single authored or multi-authored. Therefore, the RF algorithm did not consider many features as significant in this task explaining the lower overlap between it with the other algorithms on this task.

Alternatively, there was full consensus by all the three algorithms on only three features: digits, prepositions and diversity. These were the only features which appeared as significant in all the three algorithms. This overlap translated to a paltry 15.8% of the total significant features by majority voting. This was because the RF algorithm ranked only twelve (12) features as having importance values of above zero (0), thereby limiting the number of features which could be voted for by all the three algorithms. In addition, each algorithm's architecture was different with each yielding a different set of features as having importance scores of above zero (0). As such, having all the three algorithms agree on important features was dependent on each algorithms' list of important features.

Notable overlap among the three algorithms was observed with the sentence level features as opposed to the document level features. Specifically, all the ensemble algorithms had a consensus on 12 features out of the possible 22 as being significant at the sentence level. The features include total words, adverbs, word_len_gte_six, diversity, word_len_two_and_three, digits, sentence begin upper, semicolon count, long sentences, type token ratio, modals, coordinating conjunctions. The remaining ten (10) features were voted for by two algorithms; SVM and LR, SVM and RF or LR and RF. They include parenthesis count, colon count, alphabets, question sentences, nouns, prepositions, difficult words, interjections, comma count, pronouns.

In terms of percentages there was 54.5% overlap in significant features across the three algorithms. The overlap between two algorithms stood at 45.5%. The overlap between SVM and Logistic regression was 4.5% of the total number of features selected by majority voting while SVM and RF yielded an overlap of 9.1%. the greatest overlap was observed between RF and LR at 31.8%.

This study analyzed and compared important features at the sentence and document level because all the features, both sentence and document level features were ranked in the first experiment which sought to determine the best document level features. It was found out that only 35% of important document level features were determinant at the sentence level. These features included total words, digits, question sentences, adverbs, and diversity. The rest of the features were insignificant at the document level, perhaps due to the amount of the data involved. The effect of these features tends to reduce with the increase in the amount of data. For instance, semi-colon_count might be determinant in short text document such as in paragraphs or sentences, but its effect on determining writing style may reduce as the text length increases to include more paragraphs or in entire documents. Moreover, the number of nouns used could be significant in short text length and an increase or decrease in the number of nouns used could signify change in writing style. However, as the text length grows, so does the number of nouns and other factors and as such reducing the significance of nouns as a style marker in longer documents.

This study found out that syntactic features and lexical features are the best style markers in detecting writing style changes at the sentence level. This was true because out of the 22 significant sentence level features, twelve (12) were syntactic features while seven (7) were lexical features. Only two-character features ranked significant at the sentence level while there was only one (1) structural feature ranking significant at the sentence level. The findings of these studies are in line with most previous studies which opines that they could possibly be the only trusted measure of stylistic differences between works of the same or different authors (Brocardo et al., 2013; 2015). This is because it is deemed that they provide a better representation of writing styles in a much easier way because they can be normalized and quantified (Brocardo et al., 2013; 2015). However, syntactic features are language dependent and therefore were not popularly used in traditional authorship analysis studies (Alberts, 2017; Brocardo et al., 2013;

Rosso et al., 2016). State-of-the-art studies on writing style change detection employed the use of these features regardless of the need of a specific language parser (Zlatkova et al., 2018; Zuo, Zhao & Banerjee, 2019).

The results of this study laud the effectiveness of syntactic features in authorship verification and especially in writing style change detection where the document length is short. Most particularly they were among the best style markers for writing style change detection at the sentence level because of the reduced document length.

This study also found out that lexical features are good style markers for writing style change detection at the sentence level. This was evident by the fact that most word level lexical features ranked important, seven (7) features out of the twenty-two (22) sentence level features. Compared to character and structural features which had only two and one features respectively ranking as significant, the study notes the suitability of lexical features in general authorship verification tasks and more specifically in writing style change detection. Previous study show that lexical features are the most widely used features in writing style change detection a position this study supports. Literature show that lexical features have yielded promising results particularly when the document length is slightly long such as in sentence groups, paragraphs, or even entire documents (Adorno et al., 2017; Karas et al., 2017; Kuznetsov et al., 2016; Safin & Kuznetsov, 2017). However, this study show that they are effective style markers which can yield promising results even in reduced document lengths such as in a sentence i.e, they can be used to discriminate the writing styles of authors involved in writing sentences.Several studies have used different features contained in this feature set with promising results.

For instance, Deibel and Lofflad, (2021) used mean sentence length in words, mean word length or corrected type-token ratio, and pre-trained FastText embedding, with multi-layer perceptron and bidirectional LSTM for the style change detection task. Most of these features have been used in previous studies with great success, especially in studies involving short text length. For instance, counting the number of digits and parenthesis used was used in a study which sought to determine the whether a document had two or more authors and it achieved excellent performance (Sittar et al., 2016). Nath, (2019) used a combination of lexical and syntactic features to determine the number of authors in multi-authored documents and their study ranked top in the PAN@CLEF 2019 on writing style change detection.

5.3 Determining the Number of Writing Style changes in Documents

In multi-author analysis the study accomplished two tasks; classifying documents to separate single authored documents from multi-authored documents using document level features, and determining the number of style changes in documents classified as multi-authored using optimal sentence level features. Results of each task are presented below.

5.3.1 Separating Single authored Documents from Multi-authored Documents

The focus here was to classify documents as either single authored or multi-authored using ensembles of supervised learning model. Here, all the documents in the corpus were classified as either single authored or multi-authored by determining the existence of writing style changes in a document. This study presented the default parameter results and the results after hyper parameter tuning for this task. The final results of this task were the results obtained after hyper parameter tuning. The study compared results based on the feature set used. The model was trained on document level features generated separately by each algorithm, and the final optimal sentence level features generated through majority voting. Default parameter results of classifying documents into single and multi-authors are shown in table 5.9.

Table 5.9: Default Parameter results of Separating Single- from Multi-Author Documents

Feature Category	Training Accuracy	Validation Accuracy	Testing Accuracy
Random Forest best features	0.802	0.765	0.747
SVM best features	0.891	**0.870**	0.898
LR best features	**0.897**	0.841	0.906
Optimal Features (Ensemble)	0.893	0.857	0.903

From the results of Table 5.9, when the ensemble of classifiers model was trained on the best document level features obtained by Random forest, an accuracy of 0.802 was achieved on the training data, 0.765 on validation data while an accuracy of 0.747 was achieved on test data using default parameters. Logistic regression best features realized an accuracy of 0.897 on training data, 0.841 and 0.906 on validation and test data respectively. On the other hand, best features generated by SVM achieved an accuracy of 0.891 for training, 0.870 and 0.898 for validation and testing respectively. The optimal features obtained through majority voting yielded an accuracy of 0.893 for training, 0.857 for validation and testing realized an accuracy of

0.903. These results indicate that best performance was achieved when feature set generated by logistic regression was used at a testing accuracy of 0.906, followed by when the combined algorithm best features were used, at a test accuracy of 0.903. Random Forest best features produced the lowest test accuracy at 0.747.

Hyper parameter tuning was performed on the individual ensemble algorithms with a view of improving the overall performance of the ensemble. The results of the ensemble learning after hyper parameter tuning were considered as the final algorithm results for classifying documents into single authored and multi-authored and are presented in table 5.10.

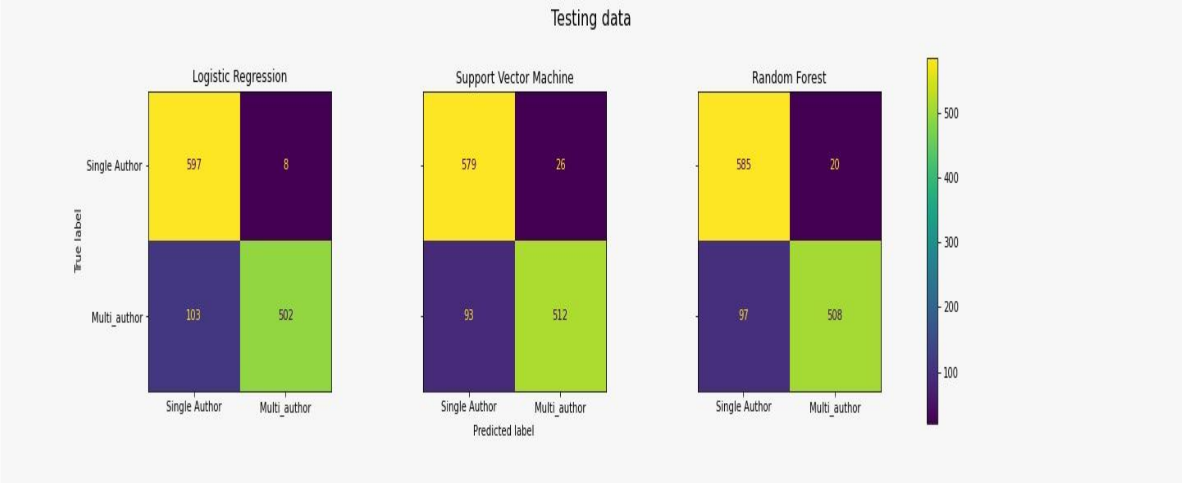
Table 5.10: Final Results of Separating Single- from Multi-Authored Documents

Feature set	Training Accuracy	Validation Accuracy	Testing Accuracy
RF feature set	0.784	0.759	0.751
SVM feature set	0.896	0.870	0.902
LR feature set	0.784	0.859	0.908
Optimal Feature set	**0.898**	**0.871**	**0.912**

According to the results of table 5.10, different feature sets produced different performance on the model. For instance, the document level features obtained using Random Forest Classifier produced an accuracy of 0.759 and 0.751 on validation and testing data respectively, while document level features according to SVM produced an accuracy of 0.870 and 0.902 for validation and testing respectively. Similarly, when the features obtained by Logistic Regression were used, an accuracy of 0.859 and 0.908 was realized with validation and testing data. The optimal feature set obtained through the ensemble by majority voting produced an accuracy of 0.871 for validation and 0.912 for testing thereby outperforming all the individual feature sets. In terms of percentages, the model achieved a performance of 75.1% using Random Forest document level features, 90.2% using SVM best features, 90.8% with Logistic Regression features and 91.2% using the ensemble optimal document level features.

Qualitatively, the classification results were also presented using a confusion matrix of four terms; True Positive, True Negative, False Positive and False Negative. True Positives indicates the documents which were correctly classified as single authored, while True negatives were multi-authored documents which were classified as multi-authored. False Positives indicated the single authored documents classified as multi-authored while False Negative were the multi-authored documents which were classified as single authors. Qualitative results of the final model performance are presented per individual algorithm followed by the results of the ensemble.

Qualitative results of the Random Forest classifier are presented in figure 5.1.



Key

True Positive	🟨
True Negative	🟪
False Positive	🟦
False Negative	🟩

Figure 5.1: Separating single authored from multi-authored using Random Forest Classifier

The results of using Logistic Regression features resulted in 597 documents were correctly classified as single authors forming the True Positives, while the True Negatives consisted of 502 documents which were correctly classified as multi-authors. Miss-classification stood at 111 documents. False Positives consisted of 8 of which were multi-authored documents classified as single authored documents and 103 single authored documents classified as multi-authored forming the False Negatives.

The results of ensemble of classifiers model trained on SVM best features resulted into 579 True Positives, 26 documents as False Positives, 93 documents as False Negatives and True Negatives of 512 documents. 579 documents were correctly classified as single authored while 93 single authored documents were classified as multi-authored by this algorithm. Similarly, 512 documents were correctly classified as multi-authored while the algorithm miss-classified 26 multi-authored documents as single authored. Comparatively, using Random Forest best features resulted in 585 True Positives and 20 False Positives. The false Negatives were 97 and True Negatives amounted to 508. This was interpreted as out of the 605 single authored documents, the ensemble of classifiers model classified 585 correctly as single authored while 20 single authored documents were classified as multi-authored by the algorithm and vice versa.

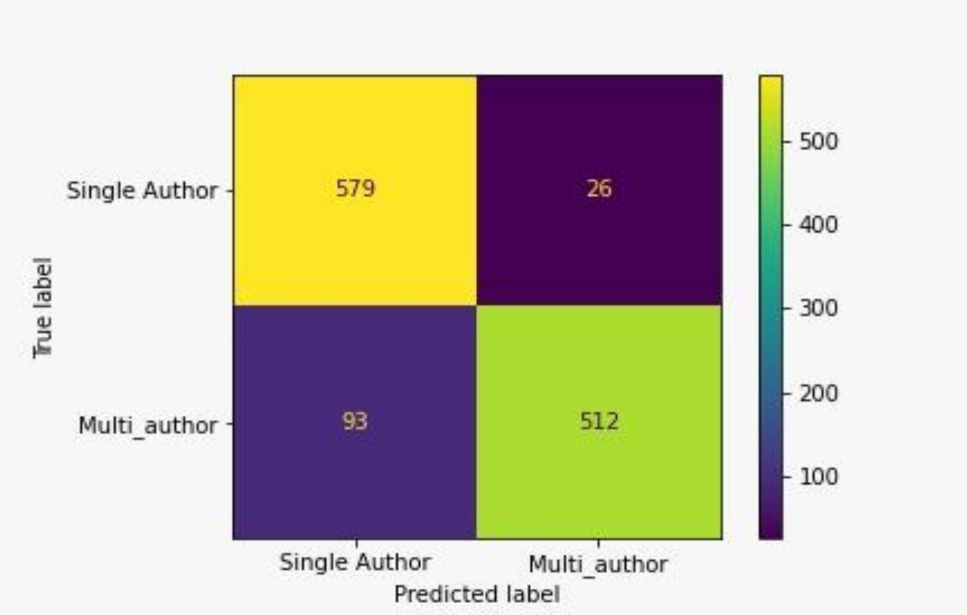


Figure 5.2: Confusion Matrix for algorithm performance using the ensemble optimal document level features

According to figure 5.2, the True Positives were 579 documents, while the True Negatives were 512 documents. False Positives amounted to 26 documents while false Negatives were 93 documents. the interpretation of the above results was given as, out of the 1210 documents in the data set, the actual labels indicted that half of the documents were single authored while the other half were multi-authored. Consequently, of the 605 single authored documents, the model classified 579 documents correctly as single authored while 512 documents were classified as multi-authored. Twenty six (26) multi-authored documents were predicted as single authored while 93 multi-authored documents were classified as single authored.

Alternative to the use of Accuracy, the study used precision and recall measures as measures of algorithm performance. Recall is the ratio of true positives to the total number of true positives and false negatives. Recall is a measure of the number of correct positive predictions made out of all positive predictions that could have been made. Precision on the other hand is a metric that quantifies the number of correct positive predictions made. F1-measure is a metric which weighs precision and recall equally and is most preferred when the data set is not balanced. The algorithm performance in terms of precision, recall and F1 score is presented in table 5.11.

Table 5.11: Precision, Recall and F1-measures of the algorithm performance

	Precision	**Recall**	**F1 score**	**Support**
Single author	0.96	0.86	0.91	605
Multi-author	0.85	0.95	0.90	605
Accuracy			0.90	1210
Macro avg	0.91	0.91	0.90	1210
Weighted avg	**0.91**	**0.91**	**0.90**	1210

From table 5.11, Precision of 0.96 and a recall of 0.86 with the single authored document as the positive class and Precision of 0.85and a recall of 0.95 with single authored being the negative class. Higher precision than recall is realized when the positive class is single authored than when it was a negative class. The higher precision meant that more false negatives were predicted by the algorithm than false positives. Specifically, more multi-authored documents were classified as single authored documents compared to the number of single authored documents which were wrongly predicted as multi-authored. However, the average values for precision and recall remained the same and was 0.91. Moreover, the algorithm realized an averaged F1 score of 0.90 for the task of separating single authored documents from multi-authored documents.

Discussion

It was observed that the use of feature sets obtained through ensemble learning achieved the best performance in training, validation and testing. Performance on the validation data set was slightly lower than those on the training data set and test set. This was partly because the data sets had very short sentences. Still the algorithm achieved acceptable performance. This study confirmed the effect of the feature set on machine learning-based algorithms. It was established that the feature set used has a significant effect on the performance of the algorithm. For instance, better performance was realized with features the feature set produced by majority voting compared to the feature sets produced by individual algorithms. Generally, the study observed that optimal features yielded best testing performance compared to the features of individual algorithms. In addition, competitive performance was achieved both for training and validation by using optimal features compared to using LR and SVM individual features.

It can also be seen that the training and test performance were almost at par, indicating that the algorithm generalizes well on these features. Improved performance obtained with ensemble learning best features was as a result of the using majority voting to select the best features from three different algorithms thereby guaranteeing the suitability of these features for writing style change detection tasks. Moreover, the three algorithms which were used in the ensemble algorithm for separating single authored documents from multi-authored documents were among the best base algorithms which have been shown to produce good performance in classification tasks. For instance, SVM and LR algorithms yield good classification accuracy in previous studies and are regarded as the go to base classifiers. Random Forest on the other hand is known for its strength in reducing biases and over fitting and therefore contributed to the better performance of the algorithm.

5.3.2 Determining the Number of Style Changes in Multi-Authored Documents

An ensemble of three clustering algorithms model was used to determine the number of style changes in documents classified as multi-authored in the first experiment. Results of training and testing and the baseline results are presented and discussed. Table 5.12 presents the evaluation results of the ensembles of clustering model on determining the number of style changes in multi-authored documents using default parameters. The focus here was to see how the different feature sets affect the algorithm performance. The OCI measure, Recall, Precision and F1 score were used to evaluate the performance of the algorithm on this task. The interpretation of the results is that the smaller the OCI value the better the performance. Specifically, as the OCI value tend to zero the better the performance. Worst performance is achieved when the OCI value is 1. Ideal performance is achieved at OCI=0.

The study presented the default parameter results and the final algorithm performance results obtained after hyper parameter tuning. Default parameter results were presented in table 5.12.

Table 5.12: Default Parameter Results of Determining the number of Style Changes in Documents based on OCI measure.

S_N	Feature set	Train_OCI	Validation_OCI	Test_OCI
1	Logistic Regression features	0.771	0.717	0.793
2	Random Forest features	0.810	0.727	0.799
3	Support vector features	0.770	0.733	0.790
4	Optimal sentence level feature set	0.804	**0.654**	**0.749**

From table 5.12, it was observed that different feature sets yielded different results: optimal sentence level features produced an OCI of 0.804, 0.749 and 0.654 for training, testing and validation. The features selected using logistic regression (Logistic) gave an OCI of 0.771, 0.793 and 0.717 for training, testing and validation. Similarly, the feature set according to the use of Random Forest yielded an OCI of 0.810, 0.799 and 0.727, while support vector machines' features produced a performance of OCI 0.770, 0.790 and 0.733 for train, test and validation respectively. The study noted that the proposed algorithm was able to generalize well using two sets of features; optimal sentence level and Random forest. This is in line with previous studies which report the superiority of ensemble methods such as random forest on the curse of dimensionality. The other two feature sets seemed to slightly over fit on the training data because they performed better in training than in testing. The proposed model performed best with the validation set which was used for hyper-parameter tuning, for all the feature sets. The best training result was achieved using support vector and logistic features at OCI of 0.770 and 0.771 respectively. The default parameter results indicate that optimal features resulted in the best performance overall for testing and validation.

Analysis of performance with test data reveal that optimal feature set was superior in terms of algorithm performance to the feature sets generated using individual algorithms. This indicates that ensemble learning methods are superior to individual algorithms in selecting the optimal feature sets for writing style change detection. In addition, it can be seen that support vector machines outperforms logistic regression and random forest in feature selection.

A number of hyper parameters were tuned in this study for individual algorithm with the assumption that they would have an effect on the ensemble results. Results of the ensemble of clustering model performance after hyper parameter tuning were presented in table 5.13.

Table 5.13: Algorithm Performance (OCI) on Determining Number of Style Changes in Multi-Authored Documents based on Different Feature Sets

Feature set	Training (OCI)	Validation (OCI)	Testing (OCI)
Optimal feature set	0.740	0.702	0.731
LR features	0.733	0.705	0.790
RF features	0.754	0.707	0.794
SVM features	0.782	0.684	0.770

From table 5.13 it can be seen that the best results were achieved with validation data for all the features sets. For instance, optimal features yielded an OCI of 0.702 and 0.740 on validation and training. Logistic regression features yielded performance of OCI 0.705 for validation and 0.733 for training. Random forest and Support Vector Machines also gave better performance on validation compared to both training and testing, of OCI 0.707 and 0.684 respectively. Overall, SVM features produced the best validation performance at OCI of 0.684 with optimal features coming second with a performance of (OCI) 0.702. Qualitatively, this study came up with a 5*5 confusion matrix to present the model performance on the task of determining the number of style changes in multi-authored documents, the results are presented in table 5.14.

Table 5.14: Results of Clustering Model Performance on Determining the number of authors.

No of authors	No. documents (True value)	No. Of documents (Predicted value)
2	**117**	**29**
3	**121**	**30**
4	**138**	**66**
5	**128**	**42**

From Table 5.14, the model performance was as follows; out of the 117 documents written by two authors, the 29 documents correctly predicted to have two-authors, Similarly, of the 121 documents which had three authors, the proposed model was able to accurately predict 30 documents to the right cluster i.e as having been written by three authors. The performance of the model on four authored documents was that of the 138 documents written by four authors, 66 documents were predicted. This class/cluster had the best algorithm performance with the highest percentage of correct predictions. Specifically, the number of correctly predicted documents outweighed all the predictions in other classes. The model predicted that 42 documents were written by five authors which formed the true predictions of the class out of 128 documents. This study concluded that the model performance on different number of style changes was actually stable although slight improvement was realized with higher number of style changes such as four. This can be attributed to the use of the optimal features and the algorithm design which included the use of three clustering algorithms and a median partition consensus function.

5.3.3 Performance of the Model on Single Features

This study compared the results of using a single feature over an expanded feature set. Three features were used having been the highest-ranking features in the three algorithms. Modal was the best feature according to Logistic Regression. Random Forest ranked mean_word_length as the best feature while Support Vector Machines ranked total_words as the best feature. Consequently, the ensemble model ranked total_words as the overall best feature. We present results of using either of the features on the ensemble clustering model in table 5.15.

Table 5.15: Algorithm Performance on Single Feature Using Ensemble of Clustering Algorithms (Insert the algorithm)

Feature	Train_OCI	Test_OCI
Modals (LR feature)	0.811	0.791
Mean_word_length (RF feature)	0.813	0.773
Total_words (SVM feature)	0.811	0.768

The ensemble of clustering algorithms was evaluated on using a single feature, the highest ranking feature in each algorithm. From the results of table 5.15, it can be seen that using modals as a feature resulted to a test performance of OCI 0.791. The use of Modal was the best feature using Logistic Regression. Mean_word_length produced an OCI of 0.773 on the test data while total_words produced the best performance on test data set at an OCI of 0.768. It is evident that the SVM algorithm’s best feature (total_words) outperformed the other two features selected using RF and LR.

When compared with the model performance using an expanded feature set, it is observed that the use of expanded feature set resulted in better performance than when a single feature was used. This finding is in line with Juola, (2006) and Nath, (2019) who recommend using more features in authorship analysis studies involving short text length.

5.4 Performance of the Ensemble Models and Benchmark Models

This study compared the performance of the ensemble models and related models on the same data set. Specifically, the work of Nath, (2019) and Zuo, Zhao & Banerjee, (2019), which sought to determine the number of authors by determining the number of writing style changes in PAN @CLEF 2019 Style change competition. The results are presented in table 5.16.

Table 5.16: Comparison of Ensemble Models' Performance versus Benchmark Models

Algorithm	Accuracy	OCI	Mean
Nath, (2019)	86.5%	0.87	0.49
Zuo, Zhao & Banerjee, (2019)	60.4%	0.80	0.40
Ensemble Models	91.2%	0.73	0.59

When the ensemble models' performance were compared with the results of the benchmark models on the same data set, it can be seen that the results of this study are superior to those of the benchmark models. The best performing model (Nath, 2019) achieved an accuracy of 86.5% while the other algorithm (Zuo, Zhao & Banerjee, 2019) obtained an accuracy of 60.4%.

Discussion

This ensemble of clustering model performed better on the test data compared to the training data and therefore is able to generalize well on unknown data. Compared to the existing model, the ensemble of clustering model was able to determine the number of style changes with improved accuracy and reduced error rate, thereby coming first compared to existing models. This was so because of the use of powerful base algorithms which are known to yield good performance. For instance, K-means algorithm which is considered the simplest yet very efficient clustering algorithm in terms of precision and recall was used. BIRCH clustering algorithm which was used together with the K-means and Gaussian in the ensemble is a more powerful clustering algorithm which is deemed to perform better than K-means with less runtime requirements. Gaussian Mixture Algorithms are able to discover complex patterns and group them into cohesive, homogeneous components that are close representatives of real patterns within the data set where K-means algorithms are limited. Therefore, combining these three algorithms in an ensemble improved greatly the performance of the algorithm.

In addition, the use of median partition-based consensus function also contributed to the performance of the models because the model was able to reduce the cases of similarity overlap. Distance-based functions are better at forming hard clusters which was advantageous to the task of determining the exact number of style changes in documents. In essence, this model was able to form distinct clusters because of the use of the three partition-based clustering algorithms and the median partition consensus function. The aim of this study was to have clear cut out clusters per document and to infer the number of authors participating in writing the document from the resultant number of distinct clusters and therefore any approach which helped to realize this was beneficial to the study.

When compared with the final model performance using an expanded feature set, it is observed that the use of expanded feature set resulted in better performance than when a single feature was used. Expanding the feature set resulted to improved performance because the model had more attributes to use to distinguish the writing style of an author from the others. Since this study adopted a case where no or limited labeled data existed, unsupervised learning algorithms were more suitable. These model perform best where there are a number of features to learn and to use to predict the unknown object. In other words, the model used in this study were unsupervised learning algorithms which required a number of features to be able to learn the writing style of an author. besides, the length of the text was reduced to a sentence, thereby requiring that more features be used to separate the works of different authors.

In general, this study noted that the model was able to identify the number of style changes in multi-authored documents with acceptable performance regardless of the number of style changes. It was observed that there was no major effect on performance by increasing the number of style changes. For instance, five-authored documents achieved better performance than four, three and two authored documents. This could be attributed to by using sentence level representation where the document is split at the sentence level and the changes in writing style is assumed to be at the sentence.

CHAPTER SIX

SUMMARY OF FINDINGS, CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

This chapter covers the summary of findings, conclusions, recommendations and future work. The rest of the chapter is organized as follows: section 6.2 covers the summary of findings, section 6.3 outlines the conclusion, and section 6.4 States the contributions made by this study while section 6.5 highlights recommendations and future work.

6.2 Summary of Findings

In this section the key findings are discussed under the various specific objectives.

6.2.1 Determining the Optimal Feature Set

The first objective of this study sought to determine the best features for separating single authored from multi-authored documents, and for determining the number of style changes in documents. The study surveyed state-of-art studies on writing style change detection and identified a number of features which have been investigated by previous studies. Experimental analysis was done on these features to reveal the best document and sentence level features which were used in the study. The following were the key findings of the study:

This study realized that a number of stylometric features exists and have been investigated for the task of writing style change detection. The most commonly used feature category for the writing style change detection is the lexical features. On the other hand, Syntactic features and character features have yielded promising results in cases where text length is small. The study analyzed both document level features and sentence level features in the PAN 2019 data set.

Feature selection is still an important exercise in writing style change detection especially where the text length is short. This is because the purity of the algorithms depends on the feature set used. For instance, using different feature set produced different results with the same algorithm. This study therefore affirms the importance of feature selection in any machine learning algorithm tasks especially in writing style change detection.

In addition, the study reports that ensembles of machine learning algorithms were the best method of selecting features since its results outperformed individual algorithm results. This is particularly true since different algorithms rank features differently with some features which rank highest in one algorithm ranking almost lowest in another algorithm. The ensemble learning has been used to solve the problem of biases and variance by combining weak learners in to strong learning in a semi-bagged fashion. Hence, the results of all the algorithms are moderated by the consensus function thereby distributing the bias and variance errors.

The findings show that colon_count, 5_gram, punctuation_count, total_words, personal_pronouns, check_uppercase, adverbs, prepositions, diversity, first_word_uppercase, 4_grams, verbs, parenthesis_count, coordinating_conjunctions, difficult_words, question_sentences, check_available_vowels, digits and adjectives were the most significant document level features.

Similarly, the most significant sentence level features were: long_sentences, word_len_two_and_three, semi-colon_count, modals, total_words, digits, coordinating_conjunctions, nouns, question_sentences, adverbs, word_len_gte_six, diversity, parenthesis_count and type_token ratio.

6.2.2 Separating Single Authored Documents from Multi-Authored Documents

This study sought to use an ensemble of machine learning algorithms algorithm to classify documents as either single authored or multi-authored. In this task, the study found out that:

This indicates that ensemble learning methods are superior to individual algorithms in selecting the best feature sets for writing style change detection and in classifying short text documents as it outperformed the other algorithms. In addition, this study also found out that for classification task, support vector machines outperformed logistic regression and random forest in feature selection.Another key finding was that Random Forest classifiers performs better for classification tasks where there are several classes than when the classification classes are well defined. For instance, it performed well when looking for sentence level features than in document level features where there were only two classes.

6.2.3 Determining the Number of Writing Style Changes in Documents

In determining the number of style changes in multi-authored documents, the study had the following findings. The ensembles of machine learning models outperformed state-of-the-art models in determining the number of authors in multi-authored documents.

Ensembles of features yields better performance than single feature analysis in writing style change detection involving short length text as opposed to uni-variate analysis where only a single feature is used. This is because the document length is too short and therefore just a single feature may not be sufficient in discriminating between the writing styles of different authors. Increasing the feature space increases the number of attributes which can be used to differentiate between works of different authors. Previous studies underscore the use of a number of features from different feature categories in improving algorithm performance. Specifically, Brocardo et al., (2015) and Nath, (2019) recommend the use of more features in studies involving short text length such as writing style change detection due to similarity overlap as the text length decreases.

The findings confirm the importance of feature selection and show that algorithm performance is as good as the features used. Although, there are no standard features sets for all the tasks of the writing style change detection and different data sets, further research in this area should be done on standardizing feature set for different tasks. Similarly, the use of ensemble learning is another factor which might have contributed to better performance of the proposed algorithm, as recommended by (Brocardo et al., 2015). Short text length could suffer from similarity overlap and over-fitting on the data if the feature set is expanded. The proposed algorithm harnesses the strengths of ensemble algorithms in mitigating over-fitting and similarity overlap.

The use of sentence level representation together with the median partition consensus function improved the performance of the writing style change detection algorithm in short text length scenarios. This is particularly so because the study did not observe degradation of performance as the number of style changes increased from three to five. Acceptable performance was achieved even when the number of style changes increased to five. This finding confirms the benefit of this algorithm on shorter text length and larger data sets.

In general, this study noted that the algorithm was able to identify the number of style changes in multi-authored documents with acceptable performance regardless of the number of authors. It was observed that there was no major effect on performance by increasing the number of authors. For instance, five-authored documents achieved better performance than four, three and two authored documents. This could be attributed to by using sentence level representation where the document is split at the sentence level and the changes in writing style is assumed to be at the sentence.

6.3 Conclusions

The main objective of this study was to develop ensembles of machine learning models for detecting writing style changes at the sentence level. Three specific objectives were investigated. The conclusion of the study is presented per objective.

6.3.1 Determining the optimal Feature Set for Writing Style Change Detection

Under this objective the study sought to determine the best feature set to use for the two tasks of separating single authored documents from multi-authored documents, and to determine the number of style changes in multi-authored documents. To achieve this objective, first the study undertook a survey on the state-of-the-art survey studies on writing style change detection and authorship verification involving short text and identified all the stylometric features which have been applied in these tasks. Then an experimental analysis of all the identified features was performed to select the best features to be applied in the study. In this regard, two experiments were performed. The first experiment was used to select the best document level features and the second experiment to select the optimal sentence level features. In both cases, an ensemble of three supervised machine learning algorithms was used to rank features based on their importance scores and majority voting used to select the best set of features. The first experiment ranked all the document level features to select the optimal feature set for the classification task.

The study concludes that the best document level features are colon_count, 5_gram, punctuation_count, total_words, personal_pronouns, check_uppercase, adverbs, prepositions, diversity, first_word_uppercase, 4_grams, verbs, parenthesis_count, coordinating_conjunctions, difficult_words, question_sentences, check_available_vowels, digits and adjectives, while the best sentence level features are long_sentences, word_len_two_and_three, semi-colon_count, modals, total_words, digits, coordinating_conjunctions, nouns, question_sentences, adverbs, word_len_gte_six, diversity, parenthesis_count and type_token ratio.

Expanding the feature set in writing style change detection significantly improves performance of the algorithms. This is so because it increases the number of attributes able to discriminate between the works of different authors. However, care should be taken to avoid over-fitting where the algorithm crams the entire data set.

6.3.2 Classifying Documents as Single-Authored and Multi-Authored

Here, the study sought to separate documents into two classes single and multi-authored. The single authored documents were dropped by the study since they were out of scope for the study while the multi-authored were used in the next experiment. To achieve this objective, the study defined two classes single-authored and multi-authored and used an ensemble algorithm of three supervised algorithms was used to classify all the documents in the data set into either of the classes. The optimal document level features from objective one was used to check for existence of multiple writing styles. Documents with a single writing style were classified as single authored while documents with more than one style were classified as multi-authored. Results of individual members of the ensemble were combined together through majority voting. This study was able to separate single authored documents from multi-authored documents with 91.2% accuracy, outperforming state-the-art studies on this data set.

6.3.3 Determining the Number of Style Changes in Multi-Authored Documents

The focus in this task was to determine the number of style changes in documents classified as multi-authored. The study developed an ensemble of three clustering algorithms to cluster together works of the same author. Each document was split into its constituent sentences with the assumption that an author writes an entire sentence. An ensemble of three clustering algorithms was used to cluster similar sentences based on the optimal sentence level features from objective 3. A consensus function based on the median partition was used to give the final results of the ensemble. The Ordinal classification Index (OCI) was used as the evaluation metric to report the errors of predicting the actual number of style changes in documents. An OCI of 0.731 was realized on the test set. The study concludes that in writing style change detection studies involving short text length, using ensembles of machine learning algorithms achieves results better performance than individual clustering algorithms. In addition, sentence level representation of features helped to improve results.

This study concludes that use sentence level representation is able to greatly improve the performance of the writing style change detection algorithms because they are able to detect even very short text contributions by authors in documents. In addition, it may contribute to consistency in performance even in larger data sets. The choice of the consensus function used to give the final output could also affect the overall performance of the algorithm. Although this study did not compare the results of using different consensus functions, the study proposed this for future work.

6.4 Contributions

This study made contributions to methodology, theory and practice.

6.4.1 Contributions to Theory

This study has made contributions to science, methodology and practice.

i. The study extended knowledge on the manner of detecting writing style changes in multi-authored documents. This was done through carrying out a systemic review of critical literature on issues of models and features that can be used in detecting changes in writing styles of different authors.

ii. Determining the optimal document level and sentence level feature sets. The optimal features are the features which yielded feature importance scores > 0. The features were determined by computing feature importance scores using an ensemble of three base algorithms. In this case, a portion of the training data was used to determine feature importance scores. The base algorithms learned from the data to give independent feature importance scores. Majority voting was then used to select features which yielded scores > 0 in more than one algorithm as the optimal features. This study determined optimal features both at the sentence and document level. The optimal document level features were colon_count, 5_gram, punctuation_count, total_words, personal_pronouns, check_uppercase, adverbs, prepositions, diversity, first_word_uppercase, 4_grams, verbs, parenthesis_count, coordinating_conjunctions, difficult_words, question_sentences, check_available_vowels, digits and adjectives, while the optimal sentence level features were; long_sentences, word_len_two_and_three, semi-colon_count, modals, total_words, digits, coordinating_conjunctions, nouns, question_sentences, adverbs, word_len_gte_six, diversity, parenthesis_count and type_token ratio. This can be found in Oloo et al., (2022a). These feature sets can be used in any writing style change detection models involving short text lengths.

iii. Designing and implementing a model; an ensemble of clustering model to determine the number of authors in documents. The model architecture consisted of three clustering algorithms; K-means, BIRCH and Gaussian Mixture Algorithms which were combined together using a median partition-based consensus function. Combining these three algorithms together in an ensemble resulted into improved performance even in scenarios where the number of authors could be possibly large. Furthermore, this is a unique architecture which has not been used in any writing style change detection as far as this study is concerned. The suitability of this model design arose from the fact that it can be applied in real world scenarios where there is limited or no labeled data yet it can still determine the number of authorship involved with improved performance compared to the existing models. Specially, K-means is a simple clustering algorithm which is able to form clusters on unseen data with improved accuracy. The BIRCH clustering algorithm is a more efficient clustering algorithm which is better than K-means because it uses a subset of the data sets instead of the full set. It is particularly useful in reducing runtime. Gaussian mixture algorithms were used because they are better algorithms in cases where the data set is highly diverse. The consensus function used allowed the final document cluster label to be determined by determining the distances between the output of each algorithm with the outputs of the other and calculating the difference in the distances between them. The final model cluster label was then given by the cluster label with the least distance between it and the rest. The use of the median-based consensus guarantees a good balance of the clusters and consistency of the results.

6.5 Recommendations and Future Work

The following recommendations were made:

i. This study performed writing style change detection at the sentence level, we propose that future work should focus on detecting writing changes which occur at the word level.

ii. The optimal document level and sentence level feature sets be tested with various models to ascertain their performance in different models and thus their effectiveness in detecting writing style changes. This is because they have only been used in one study and therefore there is need for more results from different studies to ascertain their suitability in detecting writing style changes in short text lengths should be tested with finality.

REFERENCES

Abbasi, A., & Chen, H. (2006). Visualizing authorship for identification. In Intelligence and Security Informatics: IEEE International Conference on Intelligence and Security Informatics, ISI 2006, San Diego,CA, USA, May 23-24, 2006. Proceedings 4 (pp. 60-71). Springer Berlin Heidelberg.

Abbasi, A., & Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. IEEE Intelligent Systems, 20(5), 67-75.

Abbasi, A., & Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, *26*(2), 1-29.

Adorno,G. H., Bel-Enguix, G., Vega, L. E. A., Reyes-Magaña, J. C.,(2019). MineriaUNAM at SemEval-2019 task 5: Detecting hate speech in Twitter using multiple framework.

Ahmadi, A. (2008). Multiple Cooperative Swarms for Data Clustering.

Akiva, N., & Koppel, M. (2012). Identifying distinct components of a multi-author document. Proceedings - 2012 European Intelligence and Security Informatics Conference, EISIC 2012, 205–209. https://doi.org/10.1109/EISIC.2012.16

Alberts, H. (2017). Author clustering with the aid of a simple distance measure: Notebook for PAN at CLEF 2017. CEUR Workshop Proceedings, 1866.

Alshamasi, S., & Menai, M. (2022). Ensemble-Based Clustering for Writing Style Change Detection in Multi-Authored Textual Documents. CEUR Workshop Proceedings, 3180, 2357–2374.

Alvi, F., Algafri, H., & Alqahtani, N. (2022). Style Change Detection using Discourse Markers. In *CLEF (Working Notes)* (pp. 2375-2380).

Amigud, A., Arnedo-Moreno, J., Daradoumis, T., & Guerrero-Roldan, A. E. (2017). Using learning analytics for preserving academic integrity. International Review of Research in Open and Distance Learning, 18(5), 192–210. https://doi.org/10.19173/irrodl.v18i5.3103

Anwar, W., Bajwa, I. S., & Ramzan, S. (2019). Design and implementation of a machine learning-based authorship identification algorithm. Scientific Programming, 2019. https://doi.org/10.1155/2019/9431073

Apoorva, K. A., & Sangeetha, S. (2021). Forensic Analysis of E-mail for Authorship Attribution: Research Perspective. Lecture Notes in Networks and Systems, 169 LNNS, 281–292. https://doi.org/10.1007/978-981-33-4073-2_27

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., & Shlomo Levitan, L. (2007). Stylistic text classification using functional lexical features. Journal of the American Society for Information Science and Technology, 58(6), 802–822. https://doi.org/10.1002/ASI.20553

Barlas, G., & Stamatatos, E. (2020). Cross-domain authorship attribution using pre-trained language algorithms. In IFIP Advances in Information and Communication Technology: Vol. 583 IFIP. Springer International Publishing. https://doi.org/10.1007/978-3-030-49161-1_22

Brocardo, M. L., Traore, I., & Woungang, I. (2015). Authorship verification of e-mail and tweet messages applied for continuous authentication. Journal of Computer and System Sciences, 81(8), 1429–1440. https://doi.org/10.1016/J.JCSS.2014.12.019

Brocardo, M. L., Traore, I., Saad, S., & Woungang, I. (2013). Authorship verification for short messages using stylometry. 2013 International Conference on Computer, Information and Telecommunication Systems, CITS 2013. https://doi.org/10.1109/CITS.2013.6705711

Can, F., & Patton, J. M. (2004). Change of writing style with time. Computers and the Humanities, 38(1), 61–82. https://doi.org/10.1023/B:CHUM.0000009225.28847.77

Cardoso, J. S., & Sousa, R. (2011). Measuring the performance of ordinal classification. International Journal of Pattern Recognition and Artificial Intelligence, 25(8), 1173–1195. https://doi.org/10.1142/S0218001411009093

Castro-Castro, D., Rodríguez-Losada, C. A., & Muñoz, R. (2020). Mixed Style Feature Representation and B0-maximal Clustering for Style Change Detection Notebook for PAN at CLEF 2020. In CEUR Workshop Proceedings (Vol. 2696). CEUR-WS.

Creswell, J. W., Klassen, A. C., Plano Clark, V. L., & Smith, K. C. (2011). Best practices for mixed methods research in the health sciences. *Bethesda (Maryland): National Institutes of Health*, *2013*, 541-545.

Das, P., Saha, N., & Saha, H. N. (2018). Authorship attribution of short texts using multi-layer perceptron. International Journal of Applied Pattern Recognition, 5(3), 251. https://doi.org/10.1504/ijapr.2018.10016100

Deibel, R., & Löfflad, D. (2021). Style change detection on real-world data using an LSTM-powered attribution algorithm. CEUR Workshop Proceedings, 2936, 1899–1909.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological), 39*(1), 1-22.

De Vel, O., Anderson, A., Corney, M., & Mohay, G. (2001). Mining E-mail content for author identification forensics. SIGMOD Record, 30(4), 55–64. https://doi.org/10.1145/604264.604272

Ding, S. H. H., Fung, B. C. M., Iqbal, F., & Cheung, W. K. (2016). Learning Stylometric Representations for Authorship Analysis.

Dobao, A. F. (2015). Collaborative writing in L2 classrooms. ELT Journal, 69(2), 214–216. https://doi.org/10.1093/elt/ccv001ELMANARELBOUANANI, S., & KASSOU, I. (2014). Applications, 86(12), 22–29. https://doi.org/10.5120/15038-3384Authorship Analysis Studies: A Survey. International Journal of Computer

García-Mondeja, Y., Castro-Castro, D., Lavielle-Castro, V., & Muñoz, R. (2017). Discovering author groups using a β-compact graph-based clustering. In CLEF (Working Notes), CEUR Workshop Proceedings (Vol. 1866).

Gelbukh, A. (2015). Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015 Cairo, Egypt, April 14-20, 2015 Proceedings, Part II. Lecture notes in Bioinformatics), 9042(April). https://doi.org/10.1007/978-3-319-18117-2Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence)

Ghosh, D., Khanam, A., Han, Y., & Muresan, S. (2016). Coarse-grained argumentation features for scoring persuasive essays. In 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers (pp. 549–554). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/p16-2089

Gómez-Adorno, H., Posadas-Durán, J. P., Sidorov, G., & Pinto, D. (2018). Document embeddings learned on various types of n-grams for cross-topic authorship attribution. Computing, 100, 741-756. Gómez-Adorno, H., Posadas-Durán, J. P., Sidorov, G., & Pinto, D. (2018). Document embeddings learned on various types of n-grams for cross-topic authorship attribution. Computing, 100, 741-756.

Gorman, R. (2020). Author identification of short texts using dependency treebanks without vocabulary. Digital Scholarship in the Humanities, 35(4), 812–825. https://doi.org/10.1093/LLC/FQZ07035(4), 812–825. https://doi.org/10.1093/LLC/FQZ070

Guia, M., Silva, R. R., & Bernardino, J. (2019). Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis. *KDIR*, *1*, 525-531.

Halvani, O., Lukas Graner, and Roey Regev (2017). Cross-Domain Authorship Verification Based on Topic Agnostic Features..Fraunhofer Institute for Secure Information Technology SIT Rheinstrasse 75, 64295 Darmstadt, Germany

Hosseinia, M., & Mukherjee, A. (2018). Experiments with neural networks for small and large scale authorship verification. arXiv preprint arXiv:1803.06456.

Houvardas, J., & Stamatatos, E. (2006a). N-gram feature selection for authorship identification. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4183 LNCS(May 2014), 77–86. https://doi.org/10.1007/11861461_10

Houvardas, J., & Stamatatos, E. (2006b). N-gram feature selection for authorship identification. In Artificial Intelligence: Methodology, Systems, and Applications: 12th International Conference, AIMSA 2006, Varna, Bulgaria, September 12-15, 2006. Proceedings 12 (pp. 77-86). Springer Berlin Heidelberg.

Howedi, F., Mohd, M., Aborawi, Z. A., & Jowan, S. A. (2020). Authorship Attribution of Short Historical Arabic Texts using Stylometric Features and a KNN Classifier with Limited Training Data

Howedi, F., & Mohd, M. (2014). Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data Authorship Attribution of Short Historical Arabic Texts using Stylometric Features and a KNN Classifier with Limited Training Data View project Text

Iqbal, F., Khan, L. A., Fung, B. C. M., & Debbabi, M. (2010). E-mail authorship verification for forensic investigation. Proceedings of the ACM Symposium on Applied Computing, 1591–1598. https://doi.org/10.1145/1774088.1774428

Jankowska Magdalena (2017). Author Style Analysis in Text Documents Based on Character and Word N-Grams.Dalhousie University Halifax, Nova Scotia

Jaspers, J. (2018). The transformative limits of translanguaging. Language & Communication, 58, 1-10.

Jiexun, L. I., Zheng, R., & Chen, H. (2006). From fingerprint to writeprint. In Communications of the ACM (Vol. 49, Issue 4, pp. 76–82). Association for Computing Machinery. https://doi.org/10.1145/1121949.1121951

Jung, Y. G., Kang, M. S., & Heo, J. (2014). Clustering performance comparison using K-means and expectation maximization algorithms. *Biotechnology & Biotechnological Equipment*, *28*(sup1), S44-S48.

Juola, P. (2008). Authorship attribution. Foundations and Trends® in Information Retrieval, 1(3), 233-334.

Juola, P. (2006). Authorship attribution for electronic documents. IFIP International Federation for Information Processing, 222, 119–130. https://doi.org/10.1007/0-387-36891-4_10

Karaś, D., Śpiewak, M., & Sobecki, P. (2017). OPI-JSA at CLEF 2017: Author clustering and style breach detection: Notebook for PAN at CLEF 2017. CEUR Workshop Proceedings, 1866.

Kaur, R., Singh, S., & Kumar, H. (2020). TB-CoAuth: Text based continuous authentication for detecting compromised accounts in social networks. Applied Soft Computing Journal, 97.

Kestemont, M., Tschuggnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., & Potthast, M. (2018). Overview of the author identification task at PAN-2018: cross-domain authorship attribution and style change detection. In Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al. (pp. 1-25).

Khan, J. A. (2018). A algorithm for style change detection at a glance: Notebook for PAN at CLEF 2018. CEUR Workshop Proceedings, 2125.

Kocher, M., & Savoy, J. (2017). A simple and efficient algorithm for authorship verification. Journal of the Association for Information Science and Technology, 68(1), 259-269 WHICH ONE???

Kocher, M. (2016). UniNE at CLEF 2016: Author Clustering. CEUR Workshop Proceedings, 1609, 895–902.

Koppel, M., & Winter, Y. (2014). Determining if two documents are written by the same author. Journal of the Association for Information Science and Technology, 65(1), 178-187.

Koppel, M., & Schler, J. (2004). Authorship verification as a one-class classification problem. Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004, 489–495. https://doi.org/10.1145/1015330.1015448

Kumar, S., Rajeswari, S., Srikanth, M., & Reddy, T. R. (2019). A New Approach for Authorship Verification Using Information Retrieval Features. In Lecture Notes in Networks and Systems (Vol. 74, pp. 23–29). Springer. https://doi.org/10.1007/978-981-13-7082-3_4

Kuznetsov, M. P., Motrenko, A., Kuznetsova, R., & Strijov, V. V. (2016). Methods for Intrinsic Plagiarism Detection and Author Diarization. In *CLEF (Working notes)* (pp. 912-919).

Lancashire, I., & Hirst, G. (2009). Vocabulary Changes in Agatha Christie’s Mysteries as an Indication of Dementia: A Case Study Encoding Languages View project Cybertextuality View project Vocabulary Changes in Agatha Christie’s Mysteries as an Indication of Dementia: A Case Study.

Iyer, A., Vosoughi, S.: Style Change Detection Using BERT. In: Cappellato, L., Ferro, N., Névéol, A., Eickhoff, C. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org (Sep 2020)

Mascol, C. (1888). Curves of pauline and pseudo-pauline style i. Unitarian Review, 30, 453-460.

Mendenhall, T. C. (1887). The characteristic curves of composition. Science, (214s), 237-246.

Millard, A. (2006). Book Review of The First Writing: Script Invention as History and Process, edited by Stephen D. Houston. American Journal of Archaeology, 110(3). https://doi.org/10.3764/ajaonline1103.millard

Mosteller F. and Wallace D.L. (1964) Inference and Disputed Authorship; The Federalist. Addison-Wesley Series in Behavioral Science; Quantitative Methods. Reading, Mass., Palo Alto, London, Addison-Wesley Publishing Company, Inc., XV p. 287 p

Mugenda, O.M. & Mugenda, A.G. (2003). Research Methods. Quantitative and Qualitative Approaches. Nairobi. Acts Press

NagaPrasad, S., Narsimha, V. B., Vijayapal Reddy, P., & Vinaya Babu, A. (2015). Influence of lexical, syntactic and structural features and their combination on Authorship Attribution for Telugu Tex. Procedia Computer Science, 48(C), 58–64. https://doi.org/10.1016/j.procs.2015.04.110

Nath, S. (2021). Style change detection using Siamese neural networks. In CLEF (Working Notes) (pp. 2073-2082).

Nath, S. (2019). Style Change Detection by Threshold Based and Window Merge Clustering Methods (Notebook paper) Location Aware SPARQL Query Matching (LAM) Algorithm Over Apache Spark View project Authorship Attribution View project Style Change Detection by Threshold Base.

Oloo, V. A., Otieno, C., & Wanzare, L. A. (2022a). A Literature Survey on Writing Style Change Detection Based on Machine Learning: State-Of-The-Art-Review. *vol, 70*, 15-32.

Oloo, V., Wanzare, L. D., & Otieno, C. (2022b). An Optimal Feature Set for Stylometry-based Style Change detection at Document and Sentence Level.

Ouyang, L., Zhang, Y., Liu, H., Chen, Y., & Wang, Y. (2020). Gated POS-level language algorithm for authorship verification. IJCAI International Joint Conference on Artificial Intelligence, 2021-Janua, 4025–4031. https://doi.org/10.24963/ijcai.2020/557

Pandian, A., Ragavi, R., & Ramalingam, V. V. (2020). Feature Extraction and Feature Selection process in Authorship Identification for Tamil Language. 6, 1–6. https://doi.org/10.35940/ijrte.F1001.0476S619

Pinzhakova, M., Yagel, T., & Rabinovits, J. (2021). Feature Similarity-based Regression Models for Authorship Verification. In *CLEF (Working Notes)* (pp. 2108-2117).

Potha, N., & Stamatatos, E. (2018. Intrinsic author verification using topic algorithming. ACM International Conference Proceeding Series.https://doi.org/10.1145/3200947.3201013

Potthast, M., Rangel, F., Tschuggnall, M., Stamatatos, E., Rosso, P., & Stein, B. (2017). Overview of PAN'17: author identification, author profiling, and author obfuscation. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8* (pp. 275-290). Springer International Publishing.

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing, 5*(2), 221.

Ramnial, H., Panchoo, S., & Pudaruth, S. (2016). Authorship attribution using stylometry and machine learning techniques. Advances in Intelligent Systems and Computing, 384, 113–125. https://doi.org/10.1007/978-3-319-23036-8_10

Rexha, A., Kröll, M., Ziak, H., & Kern, R. (2018). Authorship identification of documents with high content similarity. Scientometrics, 115, 223-237.

Romanov, A. S., Kurtukova, A. V., Sobolev, A. A., Shelupanov, A. A., & Fedotova, A. M. (2020). Determining the age of the author of the textbased on deep neural network algorithms. Information (Switzerland), 11(12), 1–12. https://doi.org/10.3390/info11120589

Rosso, P., Rangel, F., Potthast, M., Stamatatos, E., Tschuggnall, M., & Stein, B. (2016). Overview of PAN'16: new challenges for authorship analysis: cross-genre profiling, clustering, diarization, and obfuscation. In Experimental IR Meets Multilinguality, Multimodality, and Interaction: 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings 7 (pp. 332-350). Springer International Publishing.

Safin, K., & Ogaltsov, A. (2018). Detecting a change of style using text statistics. Working Notes of CLEF.

Safin, K. F., Kuznetsov, M. P., & Kuznetsova, M. V. E. (2017). Methods for intrinsic plagiarism detection. Informatika i Ee Primeneniya [Informatics and its Applications], 11(3), 73-79.

Sari, Y. (2018). Neural and Non-neural Approaches to Authorship Attribution.

Sari, Y., & Stevenson, M. (2016). Exploring Word Embeddings and Character N-Grams for Author Clustering. In CLEF (Working Notes) (pp. 984-991).

Schmandt-Besserat, D., & Erard, M. (2009). Origins and forms of writing. Handbook of Research on Writing: History, Society, School, Individual, Text, 30, 7–26. https://doi.org/10.4324/9781410616470-10

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*, *5*(1), 12.

Shrestha, P., Sierra, S., González, F. A., Rosso, P., Montes-Y-Gómez, M., & Solorio, T. (2017). Convolutional neural networks for authorship attribution of short texts. 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 Proceedings of Conference, 2, 669–674. https://doi.org/10.18653/v1/e17-2106

Singh, N., & Singh, D. (2012). Performance evaluation of k-means and heirarichal clustering in terms of accuracy and running time. *IJCSIT) International Journal of Computer Science and Information Technologies*, *3*(3), 4119-4121.

Sittar, A., & Ameer, I. (2018). Multi-lingual Author Profiling using Stylistic Features. In FIRE (Working Notes) (pp. 240-246).

Sittar, A., Iqbal, H. R., & Nawab, R. M. A. (2016). Author diarization using cluster-distance approach. CEUR Workshop Proceedings, 1609, 1000–1007.

Strøm, E. (2021). Multi-label Style Change Detection by Solving a Binary Classification Problem. In *CLEF (Working Notes)* (pp. 2146-2157).

Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., & Potthast, M. (2017). Overview of the author identification task at PAN-2017: style breach detection and author clustering. In Working Notes Papers of the CLEF 2017 Evaluation Labs/Cappellato, Linda [edit.]; et al. (pp. 1-22).

Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A Survey of Clustering Ensemble Algorithms. Int. J. Pattern Recognit. Artif. Intell., 25, 337-372.

Vetter, M., Sakti, S., & Nakamura, S. (2019, May). Cross-lingual speech-based Tobi label generation using bidirectional LSTM. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6620-6624). IEEE.

Weerasinghe, J., & Greenstadt, R. (2020, January). Feature vector difference based neural network and logistic regression models for authorship verification. In *CEUR workshop proceedings* (Vol. 2695).

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. Annals of Data Science, 2(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1

Zangerle, E., Mayerl, M., Potthast, M., & Stein, B. (2022). Overview of the Style Change Detection Task at PAN 2022. CEUR Workshop Proceedings, 3180, 2344–2356.

Zangerle, E., Mayerl, M., Potthast, M., & Stein, B. (2021). Overview of the style change detection task at pan.

Zangerle, E., Mayerl, M., Specht, G., Potthast, M., & Stein, B. (2020). Overview of the Style Change Detection Task at PAN 2020. CEURWorkshop Proceedings, 2696.

Zangerle, E., Tschuggnall, M., Specht, G., Stein, B., & Potthast, M. (2019). Overview of the style change detection task at pan 2019. CEUR Workshop Proceedings, 2380, 22–25.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3), 107-115.

Zhang, T., Ramakrishnan, R. & Livny, M. BIRCH: A New Data Clustering Algorithm and Its Applications. *Data Mining and Knowledge Discovery* **1**, 141–182 (1997). https://doi.org/10.1023/A:1009783824328).

Zhao, S., Xiao, Y., Ning, Y., Zhou, Y., & Zhang, D. (2021). An optimized K-means clustering for improving accuracy in traffic classification. *Wireless personal communications*, *120*, 81-93.

Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology, 57(3), 378–393. https://doi.org/10.1002/ASI.20316

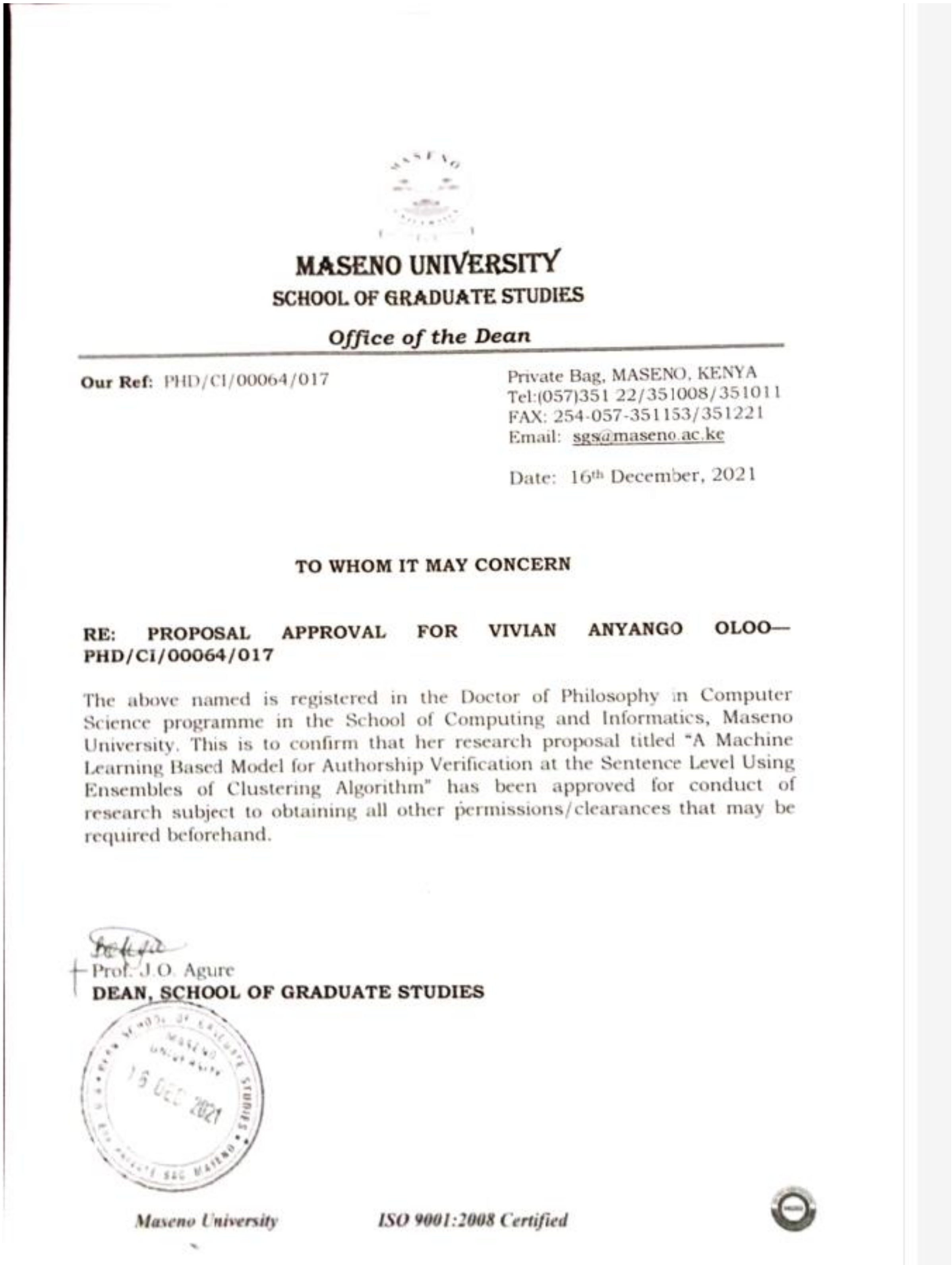
Zhou, L., & Wang, H. (n.d.). News Authorship Identification with Deep Learning.

Zlatkova, D., Kopev, D., Mitov, K., Atanasov, A., Hardalov, M., Koychev, I., & Nakov, P. (2018). An ensemble-rich multi-aspect approach for robust style change detection. In *CLEF 2018 Evaluation Labs and Workshop–Working Notes Papers, CEUR-WS. org*.

Zuo, C., Zhao, Y., & Banerjee, R. (2019). Style Change Detection with Feed-forward Neural Networks Notebook for PAN at CLEF 2019.

APPENDICES

Appendix I: SGS Approval Letter



MASENO UNIVERSITY

SCHOOL OF GRADUATE STUDIES

Office of the Dean

Our Ref: PHD/CI/00064/017

Private Bag, MASENO, KENYA
Tel:(057)351 22/351008/351011
FAX: 254-057-351153/351221
Email: sgs@maseno.ac.ke

Date: 16th December, 2021

TO WHOM IT MAY CONCERN

RE: PROPOSAL APPROVAL FOR VIVIAN ANYANGO OLOO— PHD/CI/00064/017

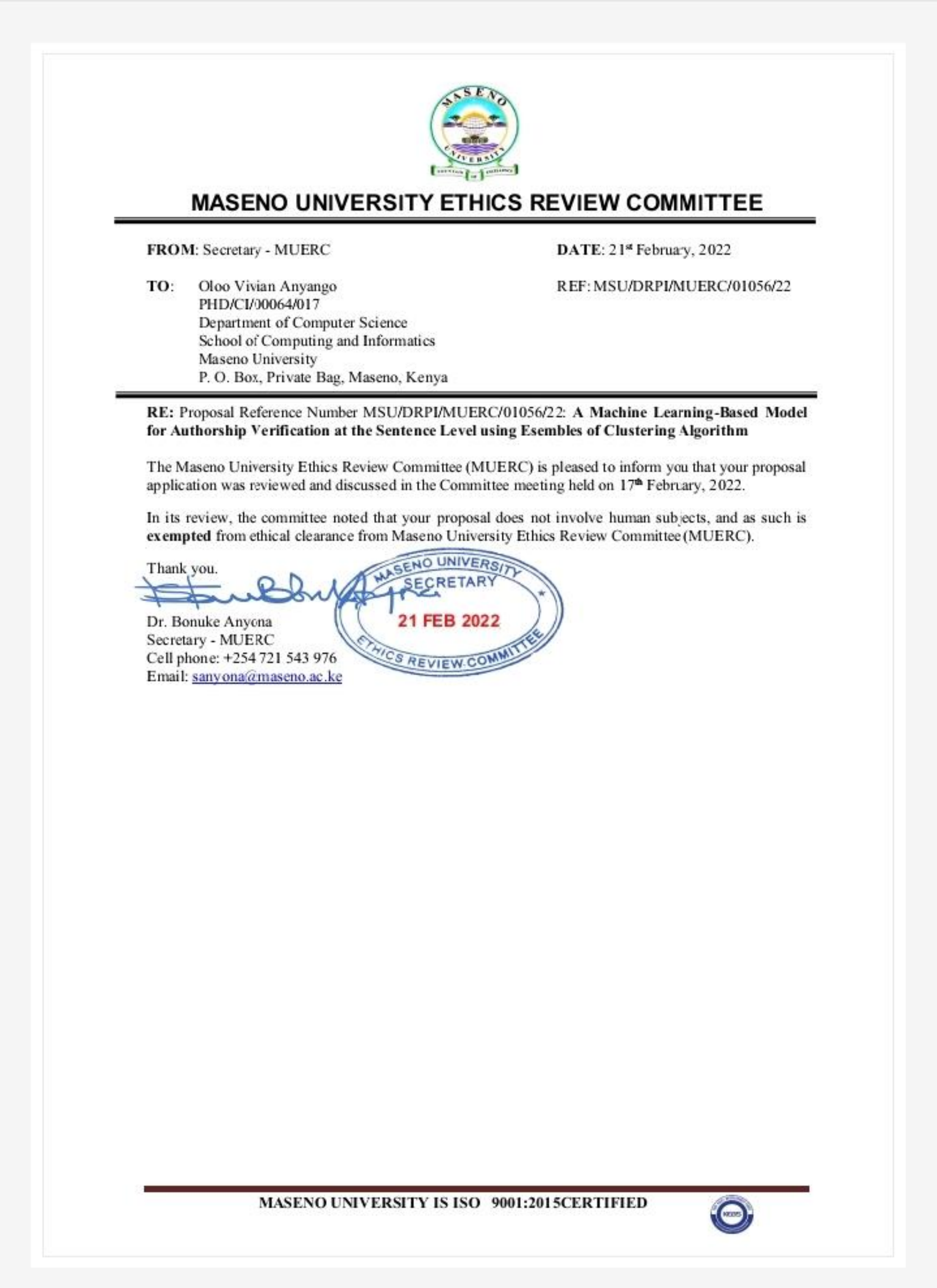
The above named is registered in the Doctor of Philosophy in Computer Science programme in the School of Computing and Informatics, Maseno University. This is to confirm that her research proposal titled "A Machine Learning Based Model for Authorship Verification at the Sentence Level Using Ensembles of Clustering Algorithm" has been approved for conduct of research subject to obtaining all other permissions/clearances that may be required beforehand.

Prof. J.O. Agure
DEAN, SCHOOL OF GRADUATE STUDIES

MASENO UNIVERSITY
16 DEC 2021
STUDIES
PRIVATE BAG MASENO

Maseno University *ISO 9001:2008 Certified*

Appendix II: MUERC Approval Letter



MASENO UNIVERSITY ETHICS REVIEW COMMITTEE

FROM: Secretary - MUERC

DATE: 21st February, 2022

TO: Oloo Vivian Anyango
PHD/CI/00064/017
Department of Computer Science
School of Computing and Informatics
Maseno University
P. O. Box, Private Bag, Maseno, Kenya

REF: MSU/DRPI/MUERC/01056/22

RE: Proposal Reference Number MSU/DRPI/MUERC/01056/22: **A Machine Learning-Based Model for Authorship Verification at the Sentence Level using Esembles of Clustering Algorithm**

The Maseno University Ethics Review Committee (MUERC) is pleased to inform you that your proposal application was reviewed and discussed in the Committee meeting held on 17th February, 2022.

In its review, the committee noted that your proposal does not involve human subjects, and as such is **exempted** from ethical clearance from Maseno University Ethics Review Committee (MUERC).

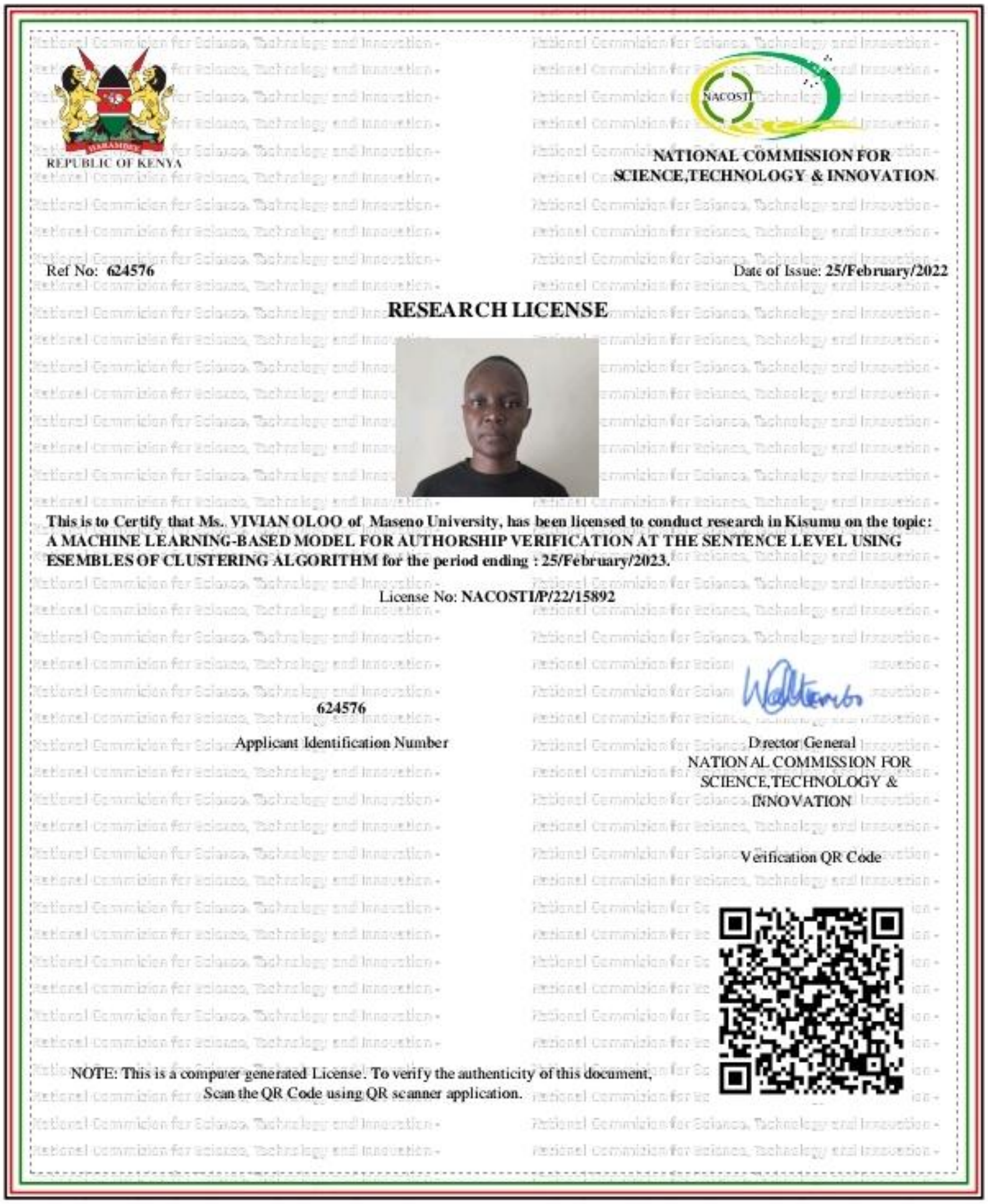
Thank you.

MASENO UNIVERSITY SECRETARY
21 FEB 2022
ETHICS REVIEW COMMITTEE

Dr. Bonuke Anyona
Secretary - MUERC
Cell phone: +254 721 543 976
Email: sanyona@maseno.ac.ke

MASENO UNIVERSITY IS ISO 9001:2015CERTIFIED

Appendix II: Research Permit



REPUBLIC OF KENYA

NACOSTI

NATIONAL COMMISSION FOR
SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **624576**

Date of Issue: **25/February/2022**

RESEARCH LICENSE

This is to Certify that Ms. VIVIAN OLOO of Maseno University, has been licensed to conduct research in Kisumu on the topic: A MACHINE LEARNING-BASED MODEL FOR AUTHORSHIP VERIFICATION AT THE SENTENCE LEVEL USING ESEMBLES OF CLUSTERING ALGORITHM for the period ending : 25/February/2023.

License No: **NACOSTI/P/22/15892**

624576

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR
SCIENCE,TECHNOLOGY &
INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document,
Scan the QR Code using QR scanner application.