IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 30, 2018, accepted August 28, 2018, date of publication September 10, 2018, date of current version September 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869198

An Effective and Scalable Framework for Authorship Attribution Query Processing

RAHEEM SARWAR[1], CHENYUN YU[2], NINAD TUNGARE[3,4] KANATIP CHITAVISUTTHIVONG[1], SUKRIT SRIRATANAWILAI[5], YAOHAI XU[3], DICKSON CHOW[3], THANAWIN RAKTHANMANON[1,5], AND SARANA NUTANONG[1]

[1]School of Information Science and Technology, Vidyasirimedhi Institute of Science and Technology, Rayong 21210,Thailand
[2]Department of Computer Science, National University of Singapore, Singapore 119077
[3]Department of Computer Science, City University of Hong Kong, Hong Kong
[4]Boston University, Boston, MA 02215, USA
[5]Department of Computer Engineering, Kasetsart University, Bangkok 10900, Thailand

Corresponding author: Chenyun Yu (yuchenyun0425@gmail.com)

This work was supported by the CityU Project under Grant 7200387 and Grant 6000511.

ABSTRACT Authorship attribution aims at identifying the original author of an anonymous text from a given set of candidate authors and has a wide range of applications. The main challenge in authorship attribution problem is that the real-world applications tend to have *hundreds of authors*, while each author may have a small number of text samples, e.g., 5–10 texts/author. As a result, building a predictive model that can accurately identify the author of an anonymous text is a challenging task. In fact, existing authorship attribution solutions based on long text focus on application scenarios, where the number of candidate authors is limited to 50. These solutions generally report a significant performance reduction as the number of authors increases. To overcome this challenge, we propose a novel data representation model that captures stylistic variations within each document, which transforms the problem of authorship attribution into a similarity search problem. Based on this data representation model, we also propose a similarity query processing technique that can effectively handle outliers. We assess the accuracy of our proposed method against the state-of-the-art authorship attribution methods using real-world data sets extracted from Project Gutenberg. Our data set contains 3000 novels from 500 authors. Experimental results from this paper show that our method significantly outperforms all competitors. Specifically, as for the closed-set and open-set authorship attribution problems, our method have achieved higher than 95% accuracy.

INDEX TERMS Query processing, large scale database, similarity search, stylometry.

I. INTRODUCTION

Authorship attribution aims at identifying the original author of an anonymous text from a set of candidate authors [1]. The authorship attribution task can be performed by comparing an anonymous text with the labeled writing samples of the candidate authors and can be formally defined as follows [2].

Definition 1 (Authorship Attribution): Given an anonymous text x, a set of candidate authors Y, and their writing samples X, identify the most likely author of x in Y by analyzing the writing samples in X and comparing them with x.

In the past few years, the applications of the authorship attribution task have increased in many areas including *intelligence agencies work*, such as, linking the intercepted messages to known enemies or terrorists [3], [4]; *criminal law*, where the main task is to identify the authors of ransome notes and harrasing letters [5]; and the *plagiarism detection* area where researchers identify whether the work submitted by a student was written by someone else [1]. An application of authorship attribution can also be found in the area of digital humanities, where issues of interest include authentication of disputed literary text. A renowned representative case of authorship attribution applications is Federalist Papers [6]. In this investigation, 12 anonymous/disputed essays were stylistically compared and analyzed against the true writing samples of James Madison and Alexander Hamilton. Nowadays, due to the increasing availability of large text repositories on the Internet, the problem of managing them is becoming more important and attracting more attention by researchers. For example, categorizing long text documents by their authors has been receiving increasing research attentions in web information management [7], information retrieval [8], and statistical natural language processing [9].

Authorship attribution problems can be categorized by the size of each text sample. For example, *short-text authorship attribution (ST-AA)* problems generally involves text samples of 600 words or less [10]–[12], which are commonly found in online social media applications [5], [10]–[12]. On the other hand, *long-text authorship attribution (LT-AA)* problems generally involves text samples containing thousands of words [2], [13]–[17], which are commonly found in digital publication and data mining applications [2], [7], [9]. This investigation is focused on LT-AA problems.

Over the past two decades, LT-AA problems have been extensively investigated by researchers in several areas such as cyber forensic, natural language processing and information retrieval. These investigations have reported a high authorship attribution accuracy (over 95%) using several kinds of style markers such as structural, syntactic, idiosyncratic and lexical ones [17]–[20]. However, applying stylometry to our large-scale LT-AA problem is a non-trivial task due to the following challenges.

- *Number of Candidate Authors:* Previous LT-AA investigations have reported a drastic drop in the performance as the number of candidate authors increases [2], [13]–[19]. Specifically, in most previous LT-AA studies reporting an accuracy of 90% or over, the number of candidate authors of the given anonymous document are limited to 50.
- *Length Variations among Writing Samples:* A variation in the document length affects the accuracy of LT-AA authorship identification [14], [21], [22].

A. PRELIMINARY CONFERENCE VERSION

In our previous investigation [1], we introduced an efficient solution to identify candidate authors using locality sensitive hashing (LSH). Specifically, we represented each document in the corpus as a set of points in a multidimensional space where each dimension corresponds to a stylometric feature. Identifying the authorship of an anonymous document Q was performed by (i) finding stylistically similar documents with respect to Q using a set-similarity measure; and (ii) applying *the probabilistic k nearest neighbor (PkNN)* classifier on identified *stylistically similar documents* to determine the most likely author. Using a corpus of 2386 novels from 136 authors, we showed that the set representation approach allowed us to handle a large number of candidate authors at a reasonable accuracy level.

For us to be able to handle a large dataset, we applied the *PkNN* technique [23]. The PkNN classifier is an instance-based learning technique and the main advantages of applying this classification technique are as follows. First, little or no training is required. Second, the learning model can make use of a complex target function. Third, there is no information loss through generalization [24]. By using the stated document representation model which represents a document in the form of a set with the PkNN classification technique, we effectively transformed the authorship attribution into a set similarity problem. This allowed us to use different set similarity functions including those which has the outlier handling techniques associated with them, such as modified Hausdorff distance, which in turn enabled us to handle a large number of candidate authors in comparison to any existing authorship attribution technique.

B. PROPOSED WORK

In this investigation, we propose the following improvements to our previously proposed solution [1]. We call our new improvement the *Stylometric Set Similarity (S3)* framework.

1) **Author Identification:** In our previous investigation, we had focused on candidate identification, *i.e., generating a small subset of candidate authors for further analysis*, rather than identifying the most likely author directly. In this investigation, *we focus on improving the accuracy of authorship identification as well.*
2) **Long Documents:** The previously proposed set comparison model allowed us to effectively compare documents with different lengths. However, we found a drastic accuracy drop when the query document is long (i.e., containing more than 120,000 words). In order to address this problem, we propose a new stylometric data representation model which organizes each document into a collection of sets, where each set has the same size.
3) **Entropy Ranking:** The new *set of sets* model effectively allows us to make multiple probabilistic predictions using multiple sets corresponding to the same document. The entropy of each prediction is calculated in order to identify those with a high prediction certainty. The final prediction result is the average of low-entropy predictions. We call this method *entropy ranking*.
4) **Open-set Author Identification:** Authorship attribution problem has two main types, namely (i) closed-set authorship attribution; and (ii) open-set authorship attribution [12]. Most existing LT-AA studies are focused on closed-set authorship attribution [2], [13]–[19]. The closed-set authorship attribution problem assumes that the original author of an anonymous document is also included in the candidate authors set. On the other hand, the open-set authorship attribution considers the possibility that none of the candidate author is the true author of the anonymous document [10], [20]. In such a case when the true author of anonymous document is not included in the candidate author set, an accurate solution should not attribute the anonymous document to any of the candidate author. In this investigation, we also show that our new *set of sets* model supports open-set authorship identification cases as well.

These modifications result in a more complete framework, a greater versatility, and an improved accuracy in comparison with our existing solution [1]. Moreover, we performed detailed experimental studies using a real-world corpus with documents written by 500 different authors to show the scalability and accuracy of our proposed method. We also introduce a comparative classification [25] method based

on *Convolutional Neural Network (CNN)* and comparative methods based on classifical machine learning algorithms into our experimental studies. Our new contributions in this paper are as follows:

- A new stylometric data representation model that can handle (i) longer documents; and (ii) a larger number of authors in comparison to the previously proposed model [1].
- An entropy ranking method which allows us to effectively make use of multiple probabilistic predictions corresponding to the same query document.
- An extension of the newly proposed solution to support open-set authorship identification.
- An expanded experimental study which includes an enlarged corpus and accuracy comparison between proposed technique and the best existing method.

The rest of the paper is structured as follows. Section II provides the literature review. The proposed solution design decisions are given in Section III. Section IV illustrates the proposed solution. Section V reports the findings obtained from our experiments. Section VI presents our concluding remarks and future works.

II. LITERATURE REVIEW

A. STYLOMETRY

Stylometry is the statistical technique that can be used to analyze the variations among the literary styles of different authors [26]. This technique has been applied to several linguistic analysis applications including author verification, author profiling, author identification and plagiarism detection.

1) STYLOMETRIC ANALYSIS TASKS

Stylometric analysis tasks are categorized into two main types, namely, authorship attribution and writing style similarity detection [17]. Each of these tasks is performed in two steps. In the first step, the stylometric features are extracted from the corpus. The second step is concerned with analyzing the feature vectors created from the first step.

The objective of authorship attribution is to compare a disputed document among labeled writing samples in order to determine the authorship of the document [2]. A renowned representative case of authorship attribution applications is Federalist Papers [6]. In this investigation, 12 anonymous/disputed essays were stylistically compared and analyzed against the true writing samples of James Madison and Alexander Hamilton. Since, the writing samples are labeled with the author names, this problem can be modeled as a supervised learning one.

The main objective of the writing style similarity detection task is to compare the query document against anonymous text samples in order to analyze the degree of similarity [17]. For example, in an anonymous online forum, one may wish to find out the number of authors by grouping comments with a similar writing style [17]. Writing style similarity detection task is an unsupervised learning problem since no class information (author labels) is available beforehand.

2) STYLOMETRIC FEATURES

Stylometric features are writing style markers that can be used to effectively discriminate the literary works of authors. Many stylometric features have been used in existing studies including syntactic, structural, idiosyncratic and lexical features:

- Syntactic features include part-of-speech *n*-grams [20], function words [6].
- Structural features are based on the organization of text, i.e., the average length of a sentence or a paragraph in terms of word count [26].
- The examples of the *Idiosyncratic features* include misspellings, grammatical mistakes and other usage anomalies.
- Lexical features include character and word-based statistical measures of lexical variations. For instance, word and character lengths, vocabulary richness [18].

3) STYLOMETRIC ANALYSIS METHODS

Stylometric analysis methods are categorized into two types, namely, supervised and unsupervised. The supervised stylometric analysis methods require class labels of text samples for classification, while unsupervised methods classify unknown object with no prior information of classes (candidate authors). Supervised methods used for stylometric analysis include neural networks, support vector machines, decision trees, radial basis function networks and nearest neighbor classification [4], [20].

The well-known unsupervised methods used for stylometric analysis include cluster analysis and principal component analysis (PCA). The ability of PCA method to reduce dimensionality across large number of features makes it suitable for stylometric analysis with large stylometric feature sets [20]. In this investigation, we first extract 56 stylometric features from the training data points. We then perform feature selection to remove redundant features and reduce the overall storage cost.

After performing feature selection analysis on 56 stylometric features, the new feature subspace consists of 40 features which can be categorized into the following types: (i) *lexical*; (ii) *syntactic*; and (iii) *structural*. Specifically, we use 16 lexical features, 22 syntactic features and 2 structural features as shown in Table 8.

4) DEEP LEARNING FOR TEXT CLASSIFICATION

In this subsection, we highlight recent developments in deep learning techniques. Recently, deep learning techniques have been extensively used to solve text classification problems. Specifically, the character-level convolutional networks report promising classification results [27], [28]. Convolutional neural networks are trained from raw character inputs. They learn the words, phrases, paragraphs from characters of the given text. However, this method require large number of

samples to show promising results. The traditional features e.g., term frequency inverse document frequency (tf-idf) and n-grams show promising results when the dataset have thousands of samples. A recent development shows that with little tuning of hyper parameters and static vectors can achieve excellent results [25].

5) AUTHORSHIP ATTRIBUTION PROBLEMS

As mentioned earlier in the introduction section, this investigation focuses on *long-text authorship attribution*. In order to provide a broader overview of different types of authorship attribution problems, we include a discussion on *short-text authorship attribution (ST-AA)* studies [10]–[12] with a large candidate author sets.

Over the past decade, considerable research attention has been dedicated to ST-AA problems which generally involve thousands of candidate authors. ST-AA techniques can be applied in several areas such as (i) social media forensics: identifying the author of a controversial post made by virtual identities on social media [10]; and (ii) *criminal law*, where the main task is to identify the authors of ransome notes and harrasing letters [5]. Generally, ST-AA studies use hundreds or thousands of features (i.e., 250,000 features) and the length of the text samples is smaller than 600 words. For example, Koppel *et al.* [12] applied their proposed technique on a corpus consisting of blog posts from 10,000 authors where the length of the test-samples is 500 words and the number of features is 250,000. Moreover, most ST-AA studies involving thousands of candidate authors [3], [11], [12]. For example, Narayanan *et al.* [3] used a corpus of blog posts from a set of 100,000 candidate authors.

Unlike ST-AA applications, the applications of LT-AA can be found in the digital publications area, where the applications of interest include authentication of disputed literary text and plagiarism detection in a student thesis [2]. Due to the availability of large text repositories on the Internet, the problem of managing them becomes more important. For example, categorizing long text documents by their authors has been receiving increasing research attentions in the areas of web information management [7], information retrieval [8], and statistical natural language processing [9].

In addition to differences in the application domains, LT-AA also differs from ST-AA in terms of the applicable stylometric features. Specifically, there are stylometric features which are applicable to LT-AA problems only. For example, vocabulary richness features used in several LT-AA problems are unstable when used with text samples shorter than 1000 words [29]–[31].

To the best of our knowledge, most previous studies on *long-text authorship attribution (LT-AA)* have used small candidate author sets in comparison to our investigation [2], [13]–[19].

Furthermore, these studies have also reported a significant drop in the accuracy of authorship attribution as the number of candidate authors increases. In addition, most LT-AA studies predict a query sample in terms of correct/incorrect classified [2], [13]–[16]. However, this is not the case with ST-AA studies reporting good precision (80% or more) [3], [12]. They allow the classifier to omit some predictions and the omitted predictions do not contribute to the overall accuracy calculations.

B. SIMILARITY SEARCH IN A MULTIDIMENSIONAL SPACE

Since our proposed solution involves the identification of stylistically similar documents using a multidimensional feature space, we briefly describe similarity search techniques in multidimensional space in the following subsections.

1) TEXTUAL SIMILARITY

Finding textual similarity can be considered as the problem of finding nearest neighbors (NNs) in a multidimensional space [32]. A naive solution, which compares each paragraph in the query document with those in the corpus, may incur prohibitive costs [33]. A more reasonable solution uses a data structure called the inverted index [32] to speed up the similarity lookup process.

A more scalable approach is that we can apply *locality sensitive hashing* (LSH) to retrieve similar paragraphs through a set of hash lookup operations. Using a technique called the *min-wise independent permutations LSH (MinHash) scheme* [34], each paragraph is represented as L hash codes computed by L different hash functions. The similarity between two paragraphs is estimated as a ratio of the number of hash code collisions and the number L of hash codes.

Comparison to Our Work: In this investigation, we represent each document as a collection of points sets in a multidimensional space. However, rather than representing each point as a Boolean vector (where each dimension corresponds to a word token), each point in our set representation is a real-valued vector where each dimension corresponds to a stylometric feature. We now discuss similarity search techniques used for a real-valued vector space.

2) INDEXING AND SIMILARITY SEARCH TECHNIQUES FOR A REAL-VALUED VECTOR SPACE.

The principle of locality sensitive hashing (LSH) was designed to perform approximate similarity search queries in multidimensional spaces [35]. Later, Datar *et al.* [36] defined an LSH function based on the p-stable distribution. They proposed the E2LSH technique to support approximate nearest neighbor (ANN) query processing. Specifically, E2LSH identifies ANN candidates as points colliding with the query point at least once. These candidates are then ranked to identify the ANN. Gan *et al.* [37] proposed *collision counting LSH (C2LSH)* technique. They have shown that the C2LSH is more suitable for range search in comparison to the E2LSH technique due to the reason that it uses the collision frequency to compute the similarity of two points in multidimensional Euclidean space [37].

Comparison to Our Work: In this investigation, we apply C2LSH technique to identify the similar data points in a multidimensional space. Specifically, we apply the C2LSH range

query to formulate a *candidate fragment pruning* method to reduce the number of stylistically similar fragments we need to consider, which in turn reduces the computational cost of our solution as shown in Section IV-C.

3) SET SIMILARITY DETECTION AND OUTLIER HANDLING TECHNIQUES

Hausdorff distance measure has been extensively used to calculate distance between two point sets in a real-valued vector space. The standard form of the well-known Hausdorff distance (SHD) can be defined as

$$H(A, B) = \max\{h(A, B), h(B, A)\},$$

where $h(A, B)$ is given by $\max_{a\in A} \min_{b\in B} d(a, b)$ and $d(a, b)$ represents the distance between two points a and b. According to definition of this set distance measure, two sets A and B are considered similar *if and only if* for every element of set A, there is at least one element in set B in proximity, and vice versa [1]. Note that $h(A, B)$ is *not* a metric distance function. This is due to the fact that, it neither holds symmetry property nor satisfies the *identity of indiscernible* principle. However, in this paper, we use the term *distance* to refer to this type of functions for conciseness. It is argued by researchers that a single outlier data point can significantly change the value of SHD [38], [39]. In order to handle the outlier sensitivity problem associated with SHD, researchers have proposed two variants of this distance measure, namely, (i) *"modified Hausdorff distance (MHD)"* [38]; and (ii) *"partial Hausdorff distance (PHD)"* [39]. As for MHD, the effect of outlier is averaged out over the minimum distances of the entire set, i.e.,

$$h_m(A, B) = \frac{1}{|A|} \sum_{a\in A} \min_{b\in B} d(a, b).$$

Dubuisson *et al.* [38] also proposed an MHD generalization in which only the top $K\%$ of the minimum distances are included in the calculation. They have shown that the MHD is effective at managing the effect of outliers in image data with noise. Huttenlocher *et al.* [39] applied a slightly different approach to deal with noise. They proposed a variant called *partial Hausdorff distance (PHD)* which treats the top $K\%$ distances as outliers and completely excludes them from computation.

The differences between Hausdorff distance variants are summarized in Figure 1. Consider two point sets A and B. As for SHD, $h(A, B)$ can be obtained by identifying the data point a in A that maximizes the minimum distance, i.e., the one at the top of the ranking. As for MHD, $h_{m,50}(A, B)$ can be obtained by identifying the top 50% distance values and then computing the average of these distances. As for PHD, $h^{75}_{p,50}(A, B)$ can be obtained by identifying the distance values between the 50^{th} percentile and 75^{th} percentile and computing the average of these distances.

Since, in this investigation, we represent each document as a collection of point sets where each point set is a real-valued vector in multidimensional space, the set distance

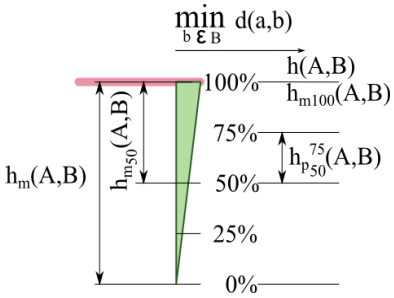


FIGURE 1. Hausdorff variants.

measures, SHD, MHD and PHD, are directly relevant to the way in which we identify stylistically similar fragments. In our experimental studies, we compare the discussed outlier management methods in the context of stylometric analysis for query processing.

C. SUMMARY

This section summarizes the main differences and advantages of our technique in comparison to the existing techniques as follows.

- The main distinction of our proposed Stylometric Set Similarity (S3) method lies in the way in which we represent each document as a collection of point sets in a multidimensional space. The proposed set representation has the following advantages.
  - First, it captures a stylistic variation within one document since each prediction is made based on multiple data points rather than just one data point.
  - Second, this representation allows us to use the set similarity measure called the Hausdorff distance and its associated outlier handling technique to identify stylistically similar documents.
- Unlike most existing authorship attribution studies, we partition this task into two parts, namely, (i) candidate generation; and (ii) author identification. The candidate generation part for which our solution generates a small subset of candidate author is performed by a set similarity search which in turn increases the accuracy of author identification part as shown in the section V.

III. SOLUTION DESIGN DECISIONS

Let us now consider the solution design decision of the *Stylometric Set Similarity (S3)* framework. We start by decomposing the authorship identification problem into two parts, *candidate identification* and *author identification*. For the candidate identification part, we generate a small subset of candidate authors from the set of all possible authors in the corpus. For the author identification part, we perform a further analysis on document samples of the candidate authors in order to identify the most likely one.

In our previous investigation [1], each document is represented as one single set of points. However, we have found that this representation format results in a drastic decrease in accuracy as the query document length increases.

In this investigation, we address this drawback by introducing another hierarchical level called the *document fragment*. Specifically, we represent each document of the corpus as a collection of fragments. We then further partition each fragment into chunks where the size of each chunk is 1,500 tokens.[1] Similar to the representation format in our previous work [1], we extract 56 features from each chunk (all the 56 stylometric features are described in Appendix A) and represent it as a 56-dimensional vector. Therefore, each *document fragment (Fragment)* corresponds to a set of points in a vector space, while each document corresponds to a collection of point sets.

Based on the stated set representation, we formulate the candidate generation problem as a set similarity problem which can be defined as follows. For a query document $\mathcal{Q}$, we decompose the given $\mathcal{Q}$ into a collection of fragments. The size of each fragment is fixed in terms of number of points in a 56 dimensional vector space. We then use Q to perform set similarity search in order to find stylistically similar fragments (SSFs) from the corpus. The SSFs authors are identified as candidate authors.

The SSFs are identified based on the set distances from the query fragment Q. In particular, we considered three set distance measures, namely, PHD, MHD and SHD. We use a set distance function (PHD, MHD or SHD) to identify the top-k SSFs that have the minimum distances with respect to the query fragment Q, where the value of k represents the desired number of candidate SSFs.

Note that, with this representation format, each fragment Q of the query document $\mathcal{Q}$ initiates an independent set similarity query. As a result, the number of candidate sets we need to consider is the same as the number of query fragments.

IV. PROPOSED SOLUTION

An overview of the proposed solution is provided in Figure 2. Our system consists of two components: *preprocessing* and *runtime query processing*. The *preprocessing* component is responsible for extracting, transforming, and loading the corpus onto the storage. The *runtime query processing* component is responsible for identifying candidate authors and performing authorship analysis based on the candidate authors identified, in order to find the most likely author.

A. PREPROCESSING

The preprocessing component of our solution aims at transforming the text documents into an easy to query format. In our previous investigation [1], we have shown that the set representation model enables us to capture stylistic variations within the same document and allows us to compare documents with different lengths. However, we found a drastic accuracy drop when the query document is long (containing more than 120,000 words). In order to address this problem,

[1]sequences of characters separated by white spaces

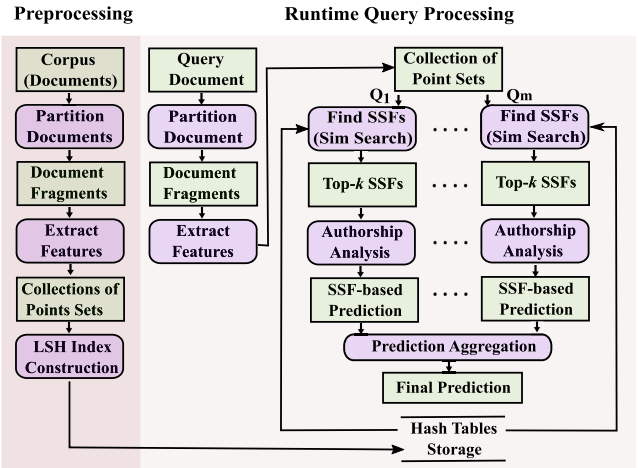


FIGURE 2. System overview: Authorship attribution based on stylometric features.

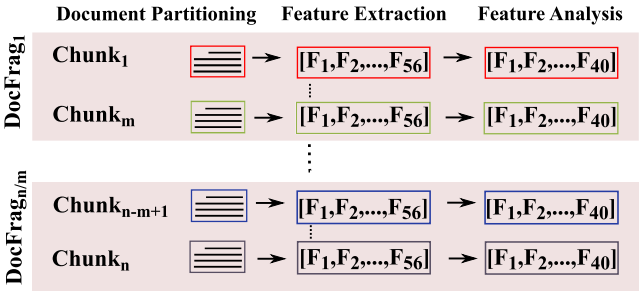


FIGURE 3. Stylometric data representation model.

we propose a new document representation model by introducing another hierarchical level, called *document fragments (Fragment)*, into our existing representation model.

1) DOCUMENT PARTITIONING

As shown in Figure 2, the first preprocessing step is to partition the documents. The document partitioning procedures are shown in Figure 3. We partition each document of the corpus into a collection of fragments. We then further partition each *fragment* into chunks where the size of each chunk is 1,500 tokens.

2) FEATURE EXTRACTION

For each chunk, we extract 56 features and represent it as a 56-dimensional vector. Descriptions of these features are given in Appendix A. Consequently, each document in the corpus corresponds to a set of fragments, while each fragment of the document in turn corresponds to a set of points in a vector space. Hence, a long document results in a large number of point sets.

Note that, we use the concept of tokens to partition a document into equal-sized chunks only. After completing document partitioning, we calculate the stylometric features from each chunk. For example, `'(word', 'word,' 'word.)'` are considered as 3 tokens. However, when

calculating word-based feature values these punctuations are removed from each token. For example, the number of distinct words in this case is 1, i.e., 'word'. On the other hand, the punctuations in each token count towards punctuation-based features. For example, the frequency of commas in this case is 2.

3) FEATURE ANALYSIS

As mentioned earlier that for each chunk, we extract 56 features. Later on, we perform feature selection using the *recursive feature elimination (RFE)* method developed by Guyon *et al.* [40] to remove redundant features and reduce the storage cost. The dimensionality reduction process consists of two main steps. The first step is concerned with subspace selection. The second step is concerned with subspace evaluation. We note that, the subspace selection step is completely unsupervised. That is, in order to identify a high variance subspace, we use the training data points only. We used this method to construct subspaces with the following numbers of dimensions: 35, 40, 45, and 50. In the evaluation step, we assess the performance of these subspaces. Specifically, in order to assess the performance of each subspace, we apply 10-fold nested cross-validation technique on the training data points and labels only. We found that the stylometric feature subspace containing 40 dimensions resulted in the best accuracy.

4) LSH INDEX CONSTRUCTION

For promoting query processing efficiency, we adopt the principle of *locality sensitive hashing* to organize data points into hash buckets where nearby data points have a greater collision probability (i.e., the probability of being assigned to the same bucket) than farther ones. In particular, we use a collision counting variant called C2LSH [37] where the distance between the two points can be estimated using the number of collisions.

Using C2LSH, all data points representing the documents are organized into L hash tables. For each hash table, we calculate the bucket ID bn of each data point using the following expression,

$$bn = \left\lfloor \frac{\vec{a}.\vec{o} + b^*}{\omega} \right\rfloor \tag{1}$$

where $\vec{o}$ denotes the vector of a data object $o \in R^d$ and $\vec{a}$ denotes a d-dimensional vector randomly chosen from a standard normal distribution, ω denotes width of the bucket and b^* is uniformly drawn from $[0, \omega]$.

5) PARAMETER SETTING

In order to improve the query efficiency by reducing the LSH lookups, we use the compound hash functions in this paper just as in the way we did in our previous work [41]. We apply the same approach to determine the LSH parameters which can be illustrated as follows. First of all, we need to specify the value of bucket width ω, query range r and approximation ratio c. Next we compute $p_1 = p(r)$, $p_2 = p(cr)$ using the following equation:

$$p(s) = \int_0^w \frac{1}{s} f\left(\frac{t}{s}\right)\left(1 - \frac{t}{w}\right) dt \tag{2}$$

where $f(x) = \frac{2}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$. In order to reduce the number of hash tables L by maximizing the value of $p_1^K - p_2^K$, we determine the value of K as follows [41]:

$$K = \left\lfloor \frac{\ln \frac{\ln p_2}{\ln p_1}}{\ln \frac{p_1}{p_2}} \right\rfloor \tag{3}$$

Note that the K value is set to 1 in the original definition of C2LSH [37]. After computing K, we can determine the values of L and α, where α denotes the collision threshold percentage. Given the false negative rate σ and false positive rate θ, we can determine the value of α as follows:

$$\alpha = \frac{z p_1^K + p_2^K}{1 + z} \tag{4}$$

where $z = \sqrt{\frac{\ln \frac{2}{\theta}}{\ln \frac{1}{\sigma}}}$, and we can determine the value L as follows.

$$L = \left\lceil \frac{\ln \frac{1}{\sigma}}{2\left(p_1^K - p_2^K\right)^z} (1 + z)^z \right\rceil \tag{5}$$

After determining the values of L and α, wet set the collision threshold T at $\alpha * L$. The final parameters used in our experiments are listed in Section V-A.

B. RUNTIME QUERY PROCESSING: SET SIMILARITY SEARCH IN MULTIDIMENSIONAL VECTOR SPACE

Given a query document $\mathcal{Q}$, we transform $\mathcal{Q}$ into a collection of fragments as described in Section IV-A. At this point, each fragment is represented as a point set. We compare each query fragment Q against all the fragments in the corpus to retrieve the top-k SSFs for further *authorship analysis* (explained in Section IV-D).

Since each query fragment Q is represented as a point set in a multidimensional space, we transform the problem of finding stylistically similar fragments into a set similarity search problem. In the next subsection, we will discuss how such an operation can be done efficiently.

C. SET SIMILARITY QUERY PROCESSING

As discussed in Section II-B.3, we consider the standard Hausdorff distance (SHD) and its variants as set similarity measures. In our previous investigation [1], we proposed a document pruning technique to reduce the computational cost of our solution. In this subsection we extend our previously proposed document pruning technique to support our newly proposed stylometric document representation model explained in Section IV-A and we call it fragment pruning technique. Specifically, our fragment pruning technique exploits the *MaxMin* nature [42] of the SHD. By exploiting

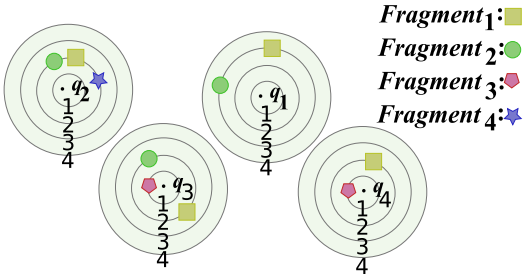


FIGURE 4. Four instances (*Fragment*$_1$, *Fragment*$_2$, *Fragment*$_3$, *Fragment*$_4$) of the range query identifying the data points in proximity of q_1, q_2, q_3, and q_4 in a query fragment Q.

TABLE 1. Lower bound calculations and Hausdorff distances calculations for the examples shown in Figure 4.

Fragment	Ranked Distances				SHD	MHD	PHD
	1_{st}	2_{nd}	3_{rd}	4_{th}	1^{st}	$[1^{st},2^{nd}]$	$[2^{nd}]$
Fragment_1	3	2	2	2	3	2.5	2
Fragment_2	> 4	3	2	2	> 4	> 3.5	3
Fragment_3	> 4	> 4	1	1	> 4	> 4	> 4
Fragment_4	> 4	> 4	> 4	2	> 4	> 4	> 4

this nature of SHD, we can avoid computing Hausdorff distance of every fragment in the corpus concerning a Q. In particular, we set the distance range threshold r for every query point q in Q to identify the candidate fragments. We then separate the identified candidate fragments in two groups. The first group contains only those candidate fragments that have a Hausdorff distance value less than or equal to r (i.e., $h(Q, \text{Fragment}) \leq r$). The second group contains rest of the candidate fragments. As for the top-k *stylistically similar fragments (SSFs)*, as long as the size of the first group of candidate fragments is larger than k value, we ignore the second group of candidate fragments. Besides that, we also generalize this fragment pruning concept to other variants of SHD known as PHD and MHD. In addition to this, we demonstrate that the task of retrieving data points closer to query point q can be accelerated greatly by using C2LSH. Besides that, we provide C2LSH error analysis for the identification of fragments in proximity to query fragment Q.

1) DISTANCE THRESHOLD-BASED FRAGMENT PRUNING TECHNIQUES FOR TOP-K SSFS PROCESSING

Fragment pruning techniques aims at reducing the quantity of fragments for which we need to evaluate the set distance with respect to the query fragment Q. As mentioned earlier, these fragment pruning techniques use a distance range threshold r to exploit *MinMax* nature of the standard Hausdorff distance (SHD) and its variants. Figure 4 illustrates how we can avoid computing SHD to every fragment from the query fragment Q by identifying those data points in a fragment which are close to every query point q in Q. Suppose that each query fragment contains 4 data points. As can be seen from Figure 4, it is guaranteed that the distance $h(Q, \text{Fragment}_1)$ is smaller than r. This is due to the reason that, for every q in Q, there is at least one data point from Fragment_1 within r. On the other hand, for Fragment_2, Fragment_3, and Fragment_4, we can see that at least one data point is missing from one of the query ranges. Consequently, the SHD between query fragment Q and *Fragment*$_2$, *Fragment*$_3$, *Fragment*$_4$ must be greater than r (i.e., $[(h(Q, \text{Fragment}_2) < r, h(Q, \text{Fragment}_3) < r$, and $h(Q, \text{Fragment}_4 < r)]$). Suppose that we want to retrieve the top-1 *stylistically similar fragment (SSF)* with respect to query fragment Q, we can directly return the Fragment_1 as a top-1 SSF and safely discard Fragment_2, Fragment_3, and Fragment_4 without computing their actual distances with respect to Q.

Let us demonstrate how we can generalize the SHD based fragment pruning concept to other variants of SHD, namely, MHD and PHD. The MHD and PHD computation process is similar to SHD computation process. Specifically, the MHD and PHD *set distance values* from Q to any fragment F can be calculated by sorting the minimum distances $\min_{p \in \text{Fragment}} d(q, p)$ for each query point q in Q. However, unlike SHD calculation process, MHD value is obtained by computing the average of maximum distances. Thus, the lower bound calculation of $h_m(Q, \text{Fragment})$ involves considering the lower bounds of all query points rather than just one.

We provide an example of Hausdorff distance computation and lower bound calculation in Table 1. This example is based on the range query process explained in Figure 4. The MHD $h_{m,50}$ shown in Table 1 is computed as the average of the 50% of the distances, such as, $h_{m,50}(Q, \text{Fragment}_1)$ is $\frac{3+2}{2}$ which is equal to 2.5 units. When one or more of the top-50% distances is a lower bound rather than an exact distance, the result from the calculation is a lower bound. For example, we can guarantee only that $h_{m,50}(Q, \text{Fragment}_2)$ is at least 3.5 units. We can also apply this principle to PHD. As shown in Table 1, PHD $h^{75}_{p,50}()$ is given as the average of distances that are in between the percentiles of 50% and 75%. For example, $h^{75}_{p,50}(Q, \text{Fragment}_1)$ is the 2^{nd} one in the example shown in Table 1. We can see that $h^{75}_{p,50}(Q, \text{Fragment}_1)$ is 2, which is the second highest distance. As for the lower bound calculation, the same method that we had derived from MHD also applies here. In this case, we can see that the lower bounds of Fragment_3 and Fragment_4 are both 4.

As for the identification of top-k SSFs, we use the concept of best first search to generate the candidate fragments result set in an incremental fashion [43]. Specifically, we first create a priority queue and populate it with the fragments. Each fragment is then ranked according to the distance lower bound. At each iteration, we retrieve the entry from the top of the priority queue. If the entry is an exact distance, we include the entry in the result set. Otherwise, we compute the exact distance and insert the entry back into the priority queue. The process terminates when the result set contains k entries.

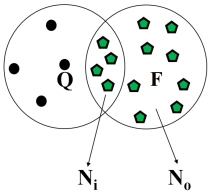


FIGURE 5. Set-level error analysis for LSH.

2) LOCALITY SENSITIVE HASHING (LSH) ERROR ANALYSIS

LSH is a popular approximation technique for solving similarity search problem in multidimensional spaces [35]. In order to identify the stylistically similar data points in a multidimensional space, we adopted C2LSH [37] proposed by Gan *et al.* They also provided an error analysis which can be used to accurately predict the recall rate in the process of range queries using C2LSH.

In this investigation, we extend the error analysis of C2LSH scheme [37] for the individual query points in order to support set distance measures (MHD, PHD and SHD). Since, our extended error analysis provided in this subsection is applicable to MHD, PHD and SHD measures, we refer to all these measures using a common term ''Hausdorff distance''.

Our error analysis is based on the false negative rate (FNR) which can be defined as follows. A fragment F is considered a false negative *if and only if* the Hausdorff distance value between this fragment F and the query fragment Q is less than the range r ($h(Q, F) \leq r$) but it is *not* identified as a near neighbor of Q. Aforesaid set-level false negatives originate due to the existence of false positives and false negatives at the point level.

Let us now evaluate the probability of a fragment F being a false negative if the Hausdorff distance value between this F and the query fragment Q is less than the range r ($h(Q, F) \leq r$). In order to evaluate that, we consider a single query point q in Q. Let N_i represent the number of data points in a fragment F that fall within the range r regarding a query point q in Q; and let N_o represent the number of data points from a fragment F that fall out of the range r. A query point q can misjudge a Fragment F as a point set which has a minimum distance greater than the range r *if and only if* the following two conditions are met. (i) All N_i data points within r must be false negatives; and (ii) all N_o data points out of the r are true positives. Let σ and θ denote the point-level FNR and point-level FPR respectively. The probability that the aforementioned two conditions are met can be calculate as $\sigma^{N_i}(1-\theta)^{N_o}$.

Now we consider the case of standard Hausdorff distance (SHD), which is the worst case scenario. Recall that, as for the SHD based fragments pruning rule for similarity search, we removed a fragment F if none of its data point appeared in the candidate result set of any q in Q. For a Q, a fragment F is considered a set-level false negative *if and only if* at least one q in Q fulfills the aforementioned conditions. Hence, the set-level FNR can be calculated as follows.

$$\text{Set-level FNR} = 1 - (1 - \sigma^{N_i}(1-\theta)^{N_o})^{|Q|} \tag{6}$$

In order to determine the values of σ and θ, we can use the following equation where δ denotes the desired set-level FNR bound.

$$\ln(1 - \sigma^{N_i}(1-\theta)^{N_o}) \leq \frac{1}{|Q|} \ln(1-\delta) \tag{7}$$

Next, we used the parameter setting methodology given in Section IV-A for experiments. Since SHD is worst case, the stated error analysis applies to other variants of SHD as well including MHD and PHD.

D. AUTHOR IDENTIFICATION

For each query fragment Q in query document $\mathcal{Q}$, the set similarity search module of our framework generates a set of SSFs. As can be seen in Figure 2, we perform an analysis on each set of SSF. The result from this step is an array of SSF-based authorship predictions. We then combine these SSF-based authorship predictions to one single prediction for the entire $\mathcal{Q}$.

In this subsection, we describe using the PkNN classifier [23] for the SSF analysis step. For the aggregation step, we propose ranking the SSF-based predictions according to the entropy in order to separate certain predictions from the less certain ones. The motivation for using PkNN includes (i) little or no model training requirement in order to perform the classification task and its ability to use the complex target functions [24]; (ii) there's no information loss through generalization [24]; (iii) it enable us to add new data at runtime and its ability to learn from a small set of samples [44]; (iv) due to its non-parametric nature, the a priori knowledge relating to probability distribution is not required for this learning method [45]; and applying the PkNN model on our document representation model (set representation) help us to transform the authorship attribution problem into a set similarity problem. As a result, we can use several set distance measures including those which have outlier handling techniques associated with them such as modified Hausdorff distance. Therefore, it enable us to handle large number of candidate authors in comparison to existing studies.

a: SSF-BASED ANALYSIS

The main advantage of using the PkNN method is that no further model training is required. Specifically, we can construct a probabilistic prediction by just analyzing the distances of the retrieved SSFs.

A straightforward way of generating a probabilistic prediction using the kNN method is to count the frequency of each class and normalize the counts by K. The resulting prediction is a *probability mass function* PMF over all classes that appear in the kNN set (i.e., SSFs in our case). Formally, we can express the (PMF) calculated by this frequency-based method

as follows:

$$p(y|x, D) = \frac{1}{K}\Sigma_{j\in neighbor(x,K,D)}I(y = y_j). \quad (8)$$

The main problem of the stated frequency-based method is that the probability associated with a class is proportional to the frequency. As a result, classes with small frequencies result in negligible probabilities [23]. To mitigate this problem, an exponential function is applied to soften the distribution. Another problem associated with the frequency-based method is that the distance of each candidate sample is ignored resulting in the need for a large K value in order to obtain reliable statistics. This problem is addressed by weighting the contribution of each sample in the kNN set using its distance [23]. By applying the exponential function β and weight function α, the resulting expression is as follows:

$$p(y|x, D, K, \beta) = \frac{exp[(\beta/K)\Sigma_{j\sim x}\alpha(x, x_j)I(y = y_j)]}{\Sigma_{y'} exp[(\beta/K)\Sigma_{j\sim x}\alpha(x, x_j)I(y' = y_j)]} \quad (9)$$

The high value of β (i.e., close to 100) results in a spiky distribution over classes. On the other hand, the low value of β uniforms the probability distribution over the classes. For the weight function α, the larger the distance from the query fragment, the less weight the fragment has. We performed several experiments to set the values of β and α parameters to get the desired accuracy.

b: PREDICTION AGGREGATION

Regardless of the learning method used in the previous step, each prediction is expressed as a PMF over a set of candidate authors. The next step is to combine all SSF-based authorship predictions to produce one single prediction for the entire query document $\mathcal{Q}$.

A straightforward method is to compute the average of all SSF-based predictions. However, all SSF-based predictions are not equally useful, e.g., highly uncertain ones. Including such predictions in the final result may damage the overall accuracy.

In this step, we use entropy as the measure to identify certain predictions. Specifically, we rank all SSF-based PMFs according to the entropy and use the most certain $\kappa\%$. The final prediction corresponds to the average PMFs of these selected predictions. This process is illustrated in Table 2 with the help of an example. For each PMF, we compute the entropy as shown in the third column and identify the top $\kappa\%$ most certain predictions (small entropy values). Assume that $\kappa = 50$. In this example, the top $\kappa\%$ most certain predictions belong to Q_2 and Q_3 as highlighted in the table. The final prediction of the entire document is computed as the average PMFs of these selected most certain $\kappa\%$ predictions, which is $[D : 0.60, E : 0.25, F : 0.15]$ in this case.

TABLE 2. Prediction aggregation with κ of 50% (*Top κ% most certain predictions).

Query Fragment	Query Fragment Prediction (PMF)	Entropy
Q_1	$[D : 0.30, E : 0.35, F : 0.35]$	1.75
$Q_2{}^*$	$[D : 0.60, E : 0.20, F : 0.20]$	0.94
$Q_3{}^*$	$[D : 0.60, E : 0.30, F : 0.10]$	0.89
Q_4	$[D : 0.34, E : 0.33, F : 0.33]$	2.13

V. PERFORMANCE EVALUATION

This section reports the findings from our experimental studies including *efficiency studies* and *accuracy studies*. As for the efficiency studies part, we provide comparison between our *LSH-based fragment pruning technique* illustrated in Section IV-C and the baseline technique which uses only the fragment pruning method. As for the accuracy study part, we compare the proposed method against the improved variation of the existing state-of-the-art authorship attribution method, AAIWE, which can handle a large candidate author set (i.e., 10,000 candidate authors) [12]. The description of the competitor (AAIW) and its improved variation AAIWE is given in the Appendix B. In addition to this, we compare our solution against classical machine learning algorithms that have been extensively used to perform the authorship attribution task, such as, *support vector machines (SVM)* [13], [16], [46]–[48], *multinomial naive bayes (MNB)* [49]–[51], as well as, the state-of-the-art deep learning technique for text classification, e.g., *convolutional neural networks (CNN)* [25], [52], [53] (cf. Section V-E).

A. EXPERIMENTAL SETUP

1) CORPUS

We extract the corpus from the online book archive, Project Gutenberg.[2] Our corpus consists of 3,000 novels from 500 authors. All these fiction novels were written in the period 1897 to 1904. Specifically, we sampled 500 authors who had written more than 2 documents. For each author, we obtained a maximum of 7 documents to keep the corpus size to a manageable level while maintaining authorship variety. We note that, the number of documents and number of authors in our corpus is significantly larger than any previous *long-text authorship attribution (LT-AA) study* [2], [13]–[16]. For example, recently published studies in top venues such as [2], [13]–[16] involve less than 20 authors and fewer than 170 documents. In comparison to our previous work [1], we increase the number of authors in our corpus from 136 to 500 authors, i.e., a 268% increase. However, the number of documents is increased slightly from 2,386 to 3,000. As a result, the average number of documents per class is reduced from 17.5 to 6 documents, thus making the classification task more difficult.

In addition to the challenge arising due to the large number of authors in the corpus, there is a significant variation

[2]https://www.gutenberg.org

among the document *lengths* i.e., 16,500 to 1,861,500 tokens. As stated earlier, each document is partitioned into fragments, where each fragment is represented by a set of 40 vectors and each vector is calculated from 1,500 tokens. As a result, the number ϕ of fragments is varied between 1 and 25 fragments.

2) EVALUATION MEASURES

We evaluated the efficiency of proposed method (S3) based on the following two measures.

(i) *Execution time*: The total execution time of the entire authorship attribution task.
(ii) *Set distance calculations*: The number of SSFs processed by the best-first search algorithm.

Measures for accuracy assessments are described as follows.

(i) *Candidate generation accuracy (CA)*: For the candidate generation accuracy, we assumed that our method was used to generate a small subset of candidate authors. Therefore, a prediction was considered correct if and only if at least one fragment of the true author is identified as a top-k SSF.
(ii) *Fragment accuracy (FA)*: A fragment-based prediction is considered correct if and only if the true author of the query fragment is identified as the most likely author.
(iii) *Authorship Attribution/Document accuracy (DA)*: An aggregated prediction of a query document is considered correct if and only if the true author is identified as the most likely author of a query document.

3) PARAMETER AND ENVIRONMENT SETTINGS

Experiments based on our proposed solution were performed on a server with the following specification: 96GB main memory, `Intel(R) Xeon(R) CPUE5-2620 v2 @ 2.10GHz` dual-processor. All algorithms were implemented in Python.

Our parameter setting contained two steps. First, we specified the set-level false negative rate (FNR) δ to 0.05, and then specified the point-level FNR σ to 0.05 and the point-level FPR θ to 0.3 based on the analysis in Section IV-C.2. After we obtained the values of σ and θ, the LSH parameters were determined by using the method illustrated in Section IV-A which can be summarized as follows.

- Specify the value of bucket width ω, query range r and approximation ratio c.
- Calculate the collision probabilities p_1 and p_2 (Equation 2).
- Determine the values of L (Equation 5) and K (Equation 3), where L denotes the number of compound hash functions and K denotes the number of projections in each compound hash function.
- Compute the collision threshold percentage α (Equation 4) and set the collision threshold T to $\alpha * L$.

The values of these parameters are given in Table 3.

TABLE 3. Parameter settings for our dataset.

c	ω	r	L	K	T
1.5	1.5	$0.15\sqrt{D}$	200	3	20

For PHD and MHD percentage ranges, we set them to [50%,75%] and (50%,100%] respectively (See Figure 1.) Although not shown here, we have tested different percentage values and ranges and those stated ones resulted in the best performance. As for the fragment size, we tested different fragment sizes. We found that the fragment size of 40 points resulted in the best performance. We also determined the best performing values for top-k and κ. They are 10 and 50% respectively.

4) EVALUATION STRATEGY

The 250 test (query) documents were from 50 different authors and each of the 50 authors had the same number of documents, i.e., 5. The 250 query documents are also organized into the following 5 different size categories: XS, S, M, L, and XL. Each author had one document in each category. The rest of the documents (2,750 of them) in the corpus were the training samples. The number of training samples for each author ranged between 4 and 6. The number of test samples per author was set at 5. Note that, the test and training samples of the same author did not come from the same novel. That is, when a novel was used for testing, it was used purely for testing.

B. EFFICIENCY STUDIES

In this phase of the investigation, we aim at evaluating the efficiency of our proposed technique using the two cost evaluation measures illustrated earlier. We compare the efficiency of our LSH-based fragment pruning technique illustrated in Section IV-C against the baseline technique which uses only the fragment pruning method illustrated in Section IV-C.1. The experimental results corresponding to the efficiency studies' are provided in Table 4. Each reported measurement shown in Table 4 is the average computed from 50 query documents $\mathcal{Q}$ with the size between 1 to 4 fragments. As can be seen, both methods had obtained a significant degree of candidate pruning. That is, almost 57% of the fragments in the corpus were considered for the case of SHD and less than 1% for all cases of MHD and PHD. The significant difference between the set distance calculation costs of SHD and two variants of Hausdorff distance, namely, MHD and PHD is due to the fact that the MHD and PHD candidate fragments were ranked according the lower bound. This allowed us to use the best-first search to control the order in which these candidates are considered. For SHD, on the other hand, the candidate fragments had no order so we needed to compute the SHD for each of them one by one. As for the execution time, Table 4 shows that, our LSH-based pruning method has provided approximately 5 times speedup as compared to the baseline.

TABLE 4. Summary of experimental results corresponding to the efficiency studies'.

Distanc	Method	Set Distance Calculations	Execution Time
SHD	Baseline	65.12%	336.64
	LSH-based pruning	56.95%	55.89
MHD	Baseline	0.21%	703.82
	LSH-based pruning	0.22%	135.97
PHD	Baseline	0.20%	703.73
	LSH-based pruning	0.20%	136.66

We also note that, as for our pruning techniques, the execution time of SHD is consistently lower than MHD and PHD. This is due to the fact that the top-k processing in case of MHD and PHD makes use of a best-first search technique on the set of candidate fragments and involve the process of lower bound calculations. Alternatively, as for SHD we could safely discard the candidate fragments which are out-of-range without a search algorithm or lower bound calculations. As can be seen that the baseline methods had significantly outperformed by our LSH-based fragment pruning methods, in the interest of brevity, we exclude their experimental results from rest of the studies.

1) EFFICIENCY: EFFECT OF VARYING THE SIZE OF QUERY SET $\mathcal{Q}$

In this subsection, we report the experimental results regarding the effect of the query set size on efficiency of our system. In order to show that the proposed solution can effectively handle the query documents of different lengths, we sample the query documents such that we can organize them according to the number of fragments, i.e., $|\mathcal{Q}|$. In particular, we organize the query documents $\mathcal{Q}$ into five groups based on their sizes $|\mathcal{Q}|$: [1, 5), [5, 10), [10, 15), [15, 20) and [20, 25).

Figure 6(a) shows the effect of $|\mathcal{Q}|$ on the efficiency cost measures. We can see that the distance calculation cost increases as the query size $|\mathcal{Q}|$ increases. This is because, an increase in the document length results in a greater number of query fragments to process. Figure 6(b) shows that the results in terms of the execution time conforms with the other cost measure.

2) EFFICIENCY: EFFECT OF VARYING THE NUMBER K OF CANDIDATE FRAGMENTS

Consider now the effect of k value on the efficiency measures. Figure 7(a) shows that varying the value of k has trivial effect on the set distance calculation cost of SHD. This is because, for SHD, we need to consider all candidates whose set distances are guaranteed to be less than r. On the other hand, for MHD and PHD, identifying the top-k involves a best-first search. A greater value of k results in a greater number of best-first search iterations. Figure 7(b) shows that

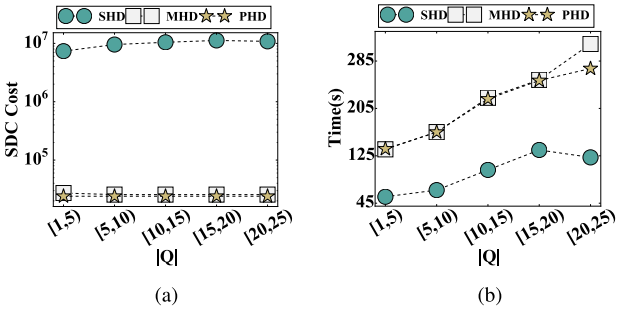


FIGURE 6. Efficiency: Effect of varying the size of query set $|\mathcal{Q}|$. (a) Set distance calculation cost. (b) Execution time.

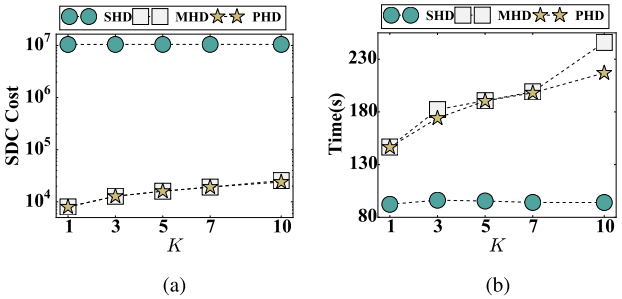


FIGURE 7. Efficiency: Effect of the top-k SSFs. (a) Set distance calculation cost. (b) Execution time.

the query execution time results conform with those of the set distance calculation cost. That is, as k increases, the query execution time also increases for MHD and PHD, while k has no effect on the query execution time of SHD.

C. ACCURACY: PROPOSED METHOD

Let us now assess the accuracy of the proposed method (S3). As mentioned earlier, each query document $\mathcal{Q}$ is decomposed into fragments where each fragment is represented as a points set Q. For each Q, we retrieve top-k SSFs. Each set of top-k SSF results in one probabilistic prediction expressed as a PMF. The final prediction for the query document $\mathcal{Q}$ is produced by combining multiple PMFs into one. In this step, we use the entropy as the measure to identify certain predictions. Specifically, we rank all SSF-based PMFs according to the entropy and use the most certain $\kappa\%$. The final prediction of the query document $\mathcal{Q}$ is produced as the average PMFs of these selected predictions with the least entropy.

Table 5 illustrates the effects of the Hausdorff distance variants on accuracy. Each result is reported as the average accuracy computed from 250 query documents. Table 5 shows that the Hausdorff variants had no effect on the candidate generation accuracy, which meant that all Hausdorff distance variations had successfully included the actual authors in the respective candidate sets. Hence, we omit the candidate generation accuracy results from the remaining subsections. We can see that, with a fragment accuracy of 95.38% and with the peak performance of 100% for the document accuracy, MHD outperformed the other Hausdorff variants. Hence, we also

TABLE 5. Accuracy: The effect of set distance measure.

Dist.	Accuracy		
	Candidate Generation	Fragment	Document
SHD	100%	75.20%	83.20%
MHD (S3)	100%	95.38%	100 %
PHD	100%	56.26%	64.00%

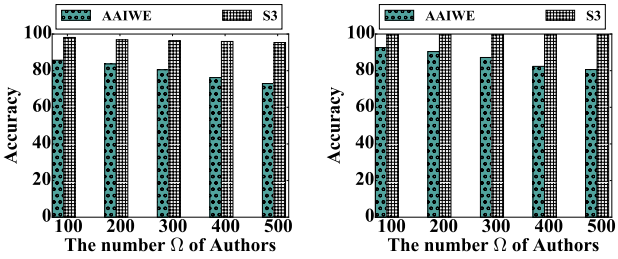


FIGURE 8. Accuracy: Effect of the number Ω of authors. (a) Fragment/Chunk Acc. (b) Document Acc.

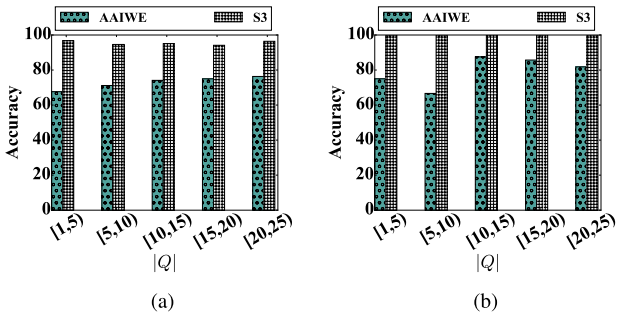


FIGURE 9. Accuracy: Effect of varying the size of query set |$\mathcal{Q}$|. (a) Fragment/Chunk Acc. (b) Document Acc.

TABLE 6. Accuracy: Effect of the chunk size.

The effect of chunk size					
Method	750	1000	1250	1500	1750
S3 (FA)	86.24%	88.73%	92.13%	95.38%	94.41%
AAIWE (FA)	62.94%	66.31%	69.92%	72.97%	73.08%
S3 (DA)	93.20%	96.80%	98.80%	100.0%	100.0%
AAIWE (DA)	63.58%	69.56%	75.16%	80.60%	80.60%

omit their results from remaining subsections. Recall that, the final prediction of a query document $\mathcal{Q}$ is produced as the average PMFs of most certain $\kappa\%$ predictions with least entropy. For the results shown in Table 5, the value of $\kappa\%$ is set to its default value which is 50%. However, in case of using all the fragment predictions of a query document $\mathcal{Q}$ (i.e., $\kappa = 100\%$), our method shows the *document accuracy* level of 95.20%.

1) ACCURACY: THE NUMBER Ω OF AUTHORS

In this study, we use 5 datasets with different numbers Ω of authors. Specifically, we varied the number Ω of candidate authors from 100 to 500. Figure 8(a) shows that increasing number of authors negatively affect the fragment accuracy and the proposed technique (S3) significantly outperforms the improved variation of existing state-of-the-art competitive technique (AAIWE). As for the document accuracy, Figure 8(b) shows that S3 is still the best performer and obtains the perfect accuracy in all cases.

2) ACCURACY: EFFECT OF VARYING THE SIZE OF QUERY SET |$\mathcal{Q}$|

Similar to Section V-B.1, in this study, we also organize the query documents $\mathcal{Q}$ into five groups according to their sizes |$\mathcal{Q}$|: [1, 5), [5, 10), [10, 15), [15, 20) and [20, 25). Figure 9(a) shows that, unlike our previously proposed method in [1], |$\mathcal{Q}$| has no significant effect on the accuracy of S3. As for the improved variation of existing state-of-the-art competitive method (AAIWE), decreasing the size of $\mathcal{Q}$ negatively effects the accuracy and this finding conforms with results reported by state-of-the-art competitive technique (AAIW) [12]. Figure 9(b) shows that S3 is still the best performer and attains perfect accuracy in all cases.

3) ACCURACY: EFFECT OF THE CHUNK SIZE

In this study, we vary the chunk size as 750, 1000, 1250, 1500 and 1750 tokens. As shown in Table 6 increasing chunk size positively affects the accuracy. However, chunk size of 1750 tokens shows only a marginal performance improvement over the chunk size of 1500 tokens. Besides that, the proposed method S3 significantly outperforms the improved variation of existing state-of-the-art competitive method AAIWE (Table 6).

D. OPEN-SET AUTHORSHIP ATTRIBUTION

The main purpose of this investigation was to devise a reliable predictive method for solving closed-set authorship attribution problems. In this study, we show how we can extend the proposed method to solve open-set authorship attribution problems. In order to conduct an open-set study, we constructed a new corpus where each author had 3 documents and there were 1500 documents. As a result, there were 500 candidate authors. There were 500 query documents where 250 of them were from the authors in the candidate set and the other 250 query documents are from non-candidate authors. All query documents were from different authors. Recall that each query fragment Q in the query document Q corresponded to one probabilistic prediction, and the overall prediction for Q was an aggregation over the fragment predictions. In the open-set study, all fragment predictions were used to compute the average PMF for $\mathcal{Q}$.

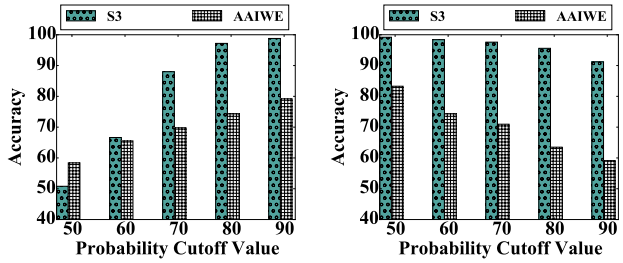


FIGURE 10. Accuracy (Document): open-set and closed-set authorship attribution. (a) Outside candidate. (b) Inside candidate.

The query document is deemed to be written by the most likely author *iff* the author had a probability larger than a predefined cutoff value. Otherwise, the prediction was considered "*uncertain*" and the document was classified as written by a non-candidate author. Based on this definition, consider now the following two cases.

- *Outside Author.* When the query document is written by a non-candidate, a prediction is considered correct *iff* no author in the prediction has a probability greater than the probability cutoff value.
- *Inside Author.* When the query document is written by an author in the candidate set, in that case a prediction is considered correct *iff* (i) the most likely author is the correct author; and (ii) its probability value is greater than the predefined cutoff.

Figure 10 displays experimental results from *outside* and *inside* author cases.

- Figure 10(a) shows that as the cutoff value increases from 50% to 90% the accuracy of identifying that the query document is written by a non-candidate author increases for both methods. This is because, as the cutoff value increases, it becomes more difficult for any author to clear the cutoff in each prediction making it easier for each prediction to be considered uncertain.
- Figure 10(b) shows that as the cutoff value increases the accuracy of identifying the author inside the candidate set decreases. This is because, since it is becoming more difficult for the most likely author to clear the probability cutoff, predictions with the correct author are more likely to be mistakenly treated as uncertain.

The experimental results also show that the proposed solution (S3) performs in most cases better than the competitor, AAIWE. We can also see that the probability cutoff value of 0.8 provides a good trade-off between the *outside inside* accuracy and *inside author* accuracy.

E. ACCURACY: COMPARISON (REDUCED CANDIDATE SET)

In this study, we compare the author identification accuracy of the following competitors which have been extensively used to perform authorship attribution task and other related text classification problems [13], [16], [25], [46]–[53].

TABLE 7. Comparison of competitive solutions.

Measure	CNN	MNB-C	MNB-T	SVM-C	SVM-T
Query Accuracy	72.98%	93.83%	90.00%	95.49%	98.28%
Training Accuracy	98.50%	98.56%	97.83%	99.99%	100%

- The *support vector machine (SVM)* classifier using two different input types *counter vector (SVM-C)* and *TF-IDF (SVM-T).*
- The *multinomial naive Bayes (MNB)* classifier using two different input types *counter vector (MNB-C)* and *TF-IDF (MNB-T).*
- The *convolutional neural networks (CNN)* technique for text classification.

Detailed descriptions of these competitive solutions are given in Appendix B. Note that the total number of authors in our corpus is 500 and none of the aforementioned competitive method is capable of handling the number of authors (classes) in this order of magnitude. *In order to compare the performance of the proposed solution against these competitors, we first use our proposed stylometric set similarity (S3) technique to identify 5 candidates and then use each of these competitors to make an authorship prediction.* In particular, each competitor is used in the *authorship analysis* part of our framework to identify the author from a small subset of candidate authors obtained using the top-k SSFs as shown in Figure 2.

Recall that one query document $\mathcal{Q}$ may contain multiple fragments where each query fragment Q results in one top-k SSF set. An authorship analysis with a competitive learning model is conducted as follows: (i) identify the candidate authors using the top-k SSFs; (ii) obtain all documents written by the identified authors; (iii) extract feature vectors from the documents as required by the competitive learning model; (iv) train the model; (v) use the model to make an SSF-based probabilistic prediction.

We tested 50 query documents $\mathcal{Q}$ in order to compare the performance of our solution against competitors classification methods. The average class size is 294 samples, where each sample corresponds to a chunk of 1,500 tokens. The number Ω of candidate authors is set to 5. This is done by ranking the authors returned from the prediction aggregation step according to the probability using our PMF aggregation method described in Section IV-D. The top Ω is selected if the number of returned authors is greater than Ω. Otherwise, all returned authors are used as candidates. Since our method always returns the true author as the most likely author, the top Ω author set is guaranteed to include the correct author for any Ω value greater than or equal to 1.

As can be seen in Table 7, SVM-T is the best performer. *However, unlike the proposed method, none of the techniques*

has the perfect query accuracy despite the fact that the size of candidate author set is significantly reduced from 500 to 5 authors. We hypothesize that the performance gap between these competitors and our method is caused by the number of samples given to train these competitive models is insufficient. This hypothesis is verified in Section Appendix C.

VI. CONCLUSION

Long-text authorship attribution (LT-AA) has been extensively studied over the past two decades by researchers in the areas of cyber forensic, web management, natural language processing and information retrieval. However, existing LT-AA studies have generally been limited to (i) text samples with similar lengths; and (ii) a small number of candidate authors. In this investigation, we have proposed a scalable solution to overcome these limitations by modeling this problem as a set similarity problem.

The main distinction of our proposed *Stylometric Set Similarity (S3)* method lies in the way in which we represent each document as a collection of point sets. The proposed set representation has the following advantages. First, it captures a stylistic variation within one document since each prediction is made based on multiple data points rather than just one data point. Second, this representation allows us to use the set similarity measure called the Hausdorff distance and its associated outlier handling technique to identify stylistically similar documents.

Our proposed solution (S3) has been evaluated using several real world datasets retrieved from Project Gutenberg. We have also compared our solution with existing state-of-the-art authorship attribution techniques. Our extensive experimental studies have shown that our method has outperformed existing state-of-the-art techniques.

One future research direction is to reduce the number of tokens required to make a reliable prediction. This can be done by applying a sliding window technique so that more text chunks can be generated from the same number of tokens or switching to a more reliable set distance to reduce the number of chunks per fragment.

APPENDIX A
STYLOMETRIC FEATURES

Our stylometric feature set is shown in Table 8. For Features 5 to 12, N denotes the total number of words and V denotes the number of distinct words. For Features 6 and 9, V_i denotes the frequency of words that occur i times. These topic-independent stylometric features [6], [18], [20], [54], [55], can be categorized into the following types: (i) *lexical*; (ii) *syntactic*; and (iii) *structural*.

TABLE 8. List of Stylometric Features(* Features selected after a feature reduction analysis). Descriptions of these features are given in Appendix A.

Lexical Features	
1. N: Total #words	*2. V: Total #distinct words
*3. Average word length	*4. S.D. of word lengths
*5. $\frac{V}{N}$	*6. $VR(K) = \frac{10^4(\sum i^2 V_i - N)}{N^2}$
7. $VR(R) = \frac{V}{\sqrt{N}}$	*8. $VR(C) = \frac{\log V}{\log N}$
9. $VR(H) = \frac{(100 \log N)}{(1-V_1)/V}$	10. $VR(S) = \frac{V_2}{V}$
*11. $VR(k) = \frac{logV}{\log(\log N)}$	12. $VR(LN) = \frac{(1-V^2)}{V^2(\log N)}$
*13. Entropy of word freq. ditro.	*14. Total number of chars
*15. Freq. of alpha chars	*16. Freq. of uppercase chars
17. Freq. of lowercase chars	*18. Freq. of numeric chars
*19. Freq. of special chars	20. Freq. of white spaces
*21. Freq. of punctuations	22. Alpha char ratio
*23. Uppercase char ratio	24. Lowercase char ration
*25. Numeric char ratio	*26. Special char ratio
27. White spaces ratio	
Syntactic Features	
*28. Freq. of nouns	*29. Freq. of proper nouns
*30. Freq. of pronouns	*31. Freq. of ordinal adjs.
*32. Freq. of comparative adjs.	*33. Freq. of superlative adjs.
*34. Freq. of advs.	*35. Freq. of comparative advs.
36. Freq. of superlative advs.	*37. Freq. of modal auxiliaries
38. Freq. of bases form verbs	39. Freq. of past verbs
*40. Freq. of present part. verbs	41. Freq. of past part. verbs
*42. Freq. of particles	*43. Freq. of wh-words
*44. Freq. of conjunctions	*45. Freq. of numerical words
*46. Freq. of determiners	*47. Freq. of existential theres
*48. Freq. of existential to	*49. Freq. of prepositions
*50. Freq. of genitive markers	*51. Freq. of quotations
*52. Freq. of commas	53. Freq. of terminators
54. Freq. of symbols	
Structural Features	
*55. Total number of sentence	*56. Avg. #words per sentence

APPENDIX B
DESCRIPTIONS OF COMPETITIVE METHODS

A. AAIW AND ITS IMPROVED VARIATION (AAIWE)

We compare the performance of our proposed method, S3, against the state-of-the-art existing authorship attribution method that can handle a large candidate author set (i.e., 10,000 authors) called AAIW [12]. AAIW represents each text sample as a vector containing the respective frequencies of each space-free character 4-gram. It uses the cosine similarity as a proximity measure and returns the author whose known writing sample is more similar to the test sample. For each test sample, 100 predictions are made using different subsets of features randomly selected from a set of 2.5×10^5 features. Each candidate author is assigned a score which is calculated as the proportion of the number of times that candidate author is the top match. The author obtaining the maximum overall score is considered the most likely author. In addition to the making an authorship prediction, AAIW also allows the attribution to omit a prediction. Specifically, a threshold value of 0.90 is chosen to exclude predictions in which the mostly likely author scores less than 0.90. Such predictions are labeled as ''Don't Know'' and are excluded from the accuracy assessment.

TABLE 9. Fragment Accuracy: Comparison of baseline technique with its improved version.

The effect of the chunk size					
Method	750	1000	1250	1500	1750
AAIWE	62.94%	66.31%	69.92%	72.97%	73.08%
AAIW	46.39%	53.49%	59.11%	63.47%	63.81%

Since AAIW uses the *k nearest neighbor (kNN)* classifier, the method is also applicable to the *probabilistic kNN (PkNN)* method. In this investigation, we have also shown that we can apply the prediction aggregation method discussed in Section IV-D to improve the accuracy of AAIW. Specifically, we apply the P*k*NN weighting function (Eq. 9) to the cosine similarity search results for each test sample. We then apply the entropy-ranking prediction aggregation method illustrated in Table 2. We call this improvement *AAIW-Entropy*, which is abbreviated to *AAIWE* for conciseness. We conducted an experimental study using the same corpus as the main experiment to compare AAIWE to its original version AAIW. The results presented in Table 9 show that AAIWE outperforms the baseline technique AAIW. As a result, AAIWE is used as our competitor in the experimental studies (Section V).

B. SUPPORT VECTOR MACHINES (SVM)

The machine learning method, SVM, has been applied to a wide variety of text categorization problems including authorship attribution [13], [16], [46]–[48]. In this investigation, SVM is used with two different input types: *count vectors* and *TF-IDF*, and call these two variations SVM-C and SVM-T, respectively. In our experimental studies, we used the Scikit-learn machine learning library to implement the two SVM solutions. The parameters used construct our SVM classifiers are given in Table 10.

C. MULTINOMIAL NAIVE BAYES (MNB)

The multinomial Naive Bayes (MNB) classifier has also reported competitive results for authorship attribution and text classification problems [49]–[51], [56]. Similar to the SVM method, we apply the MNB method to two different input types: *count vectors* and *TF-IDF*, and call these two variations MNB-C and MNB-T, respectively. In our experimental studies, we used the Scikit-learn machine learning library to implement the two MNB solutions. The parameters used to construct our MNB classifiers are given in Table 10.

D. CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNN have been extensively used to solve text classification and authorship attribution problems [25], [52], [53]. Convolutional neural networks are trained from raw character inputs. They learn the words, phrases, paragraphs from characters of the given text. However, this method require large number of samples to show promising results. The traditional features e.g., n-grams, term frequency inverse document frequency (tf-idf) show promising results when the dataset have thousands of samples. A recent development shows that with little tuning of hyper parameters and static vectors can achieve excellent results [25].

In this investigation, we adapted a state-of-the-art CNN architecture proposed by Kim [25]. The learning model uses word *n*-gram features as input for a CNN, which is in turn connected to a fully connected neural network with 3 layers. Given the constraint of corpus size, we decided to use a pre-trained word embedding technique [57] to convert each input word into a multidimensional vector. The characteristics of our adapted CNN architecture are displayed in Table 11.

TABLE 10. Parameters for the SVM and MNB solutions.

Parameter	Value
Features	word n-gram; $n = [1-4]$
Max. No. of Features	40,000
Stop Words	None
Lowercase	False
Use_idf	True
Smooth_idf	True
Sublinear_tf	True
Probability	True

TABLE 11. Characteristics of the CNN method.

Embedding Layer			
Embedding	Input	Vector	Method
Vectorization	Tokenized Words	200D vectors	GloVe [58]
Convolutional Layers			
Layer	Output Features	Convolution Kernel	Pool
X a	256	3	3
X b	256	4	3
Fully Connected Layers			
Layer	Number of Output Features		Dropout
2	Merged o/p from the prev. layer		0
3	256		0.5
4	Number of authors		-

APPENDIX C
ACCURACY STUDIES: LARGE NUMBER OF SAMPLES

We verify the hypothesis given in the experimental studies (Section V-E) regarding the insufficient number of samples. In order to obtain a larger number of samples for this study,

TABLE 12. The effect of the number of samples.

	Number of Samples		
Method	320	1600	3200
CNN	68.12 %	85.56 %	89.05%
MNB-C	94.68%	97.12%	96.65%
MNB-T	93.12 %	96.31%	96.65%
SVM-C	96.87%	98.75%	98.65%
SVM-T	98.43 %	99.68%	99.96%

TABLE 13. The effect of the number Ω of authors.

	Number Ω of authors			
Method	2	3	4	5
CNN	96.30%	95.20	91.65%	89.05%
MNB-C	96.95%	96.87%	96.25%	96.65%
MNB-T	97.03%	97.18%	96.17%	96.65%
SVM-C	99.68%	99.42%	99.25%	98.65%
SVM-T	99.92%	100%	100%	99.96%

we identify a set of 5 authors with at least 3,200 samples each. We conduct two studies: (i) the effect of the number Ω of authors; (ii) the effect of the number of samples. Each accuracy value reported is the result from a 4-fold cross validation. That is, 75% of the sample set is used for training.

A. THE EFFECT OF THE NUMBER OF SAMPLES

We vary the number of samples from 320 to 3,200 samples while keeping the number Ω of candidate authors to the default value of 5. Table 12 shows that for all methods, the number of samples has a positive correlation with the accuracy. The CNN method has the most drastic accuracy increase, i.e., from 68.12 to 88.39, which is similar to the results reported by Kim [25]. This result conforms with our hypothesis regarding the insufficient number of samples discussed in Section V-E.

B. THE EFFECT OF THE NUMBER Ω OF AUTHORS

Table 13 shows a drastic accuracy drop for the CNN method as Ω is increases from 2 to 5, while fixing the number of samples to 3,200 samples. For all classical ML methods, the varying Ω from 1 to 5 has little or no effect on the accuracy.

REFERENCES

[1] S. Nutanong, C. Yu, R. Sarwar, P. Xu, and D. Chow, "A scalable framework for stylometric analysis query processing," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1125–1130.
[2] H. Ramnial, S. Panchoo, and S. Pudaruth, "Authorship attribution using stylometry and machine learning techniques," in *Intelligent Systems Technologies and Applications*. Cham, Switzerland: Springer, 2016, pp. 113–125.
[3] A. Narayanan *et al.*, "On the feasibility of Internet-scale author identification," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2012, pp. 300–314.
[4] R. Sarwar, Q. Li, T. Rakthanmanon, and S. Nutanong, "A scalable framework for cross-lingual authorship identification," *Inf. Sci.*, vol. 465, pp. 323–339, Oct. 2018.
[5] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "A unified data mining solution for authorship analysis in anonymous textual communications," *Inf. Sci.*, vol. 231, no. 9, pp. 98–112, 2013.
[6] D. L. Mosteller and F. Wallace, *Inference and Disputed Authorship, The Federalist*. Boston, MA, USA: Addison-Wesley, 1964.
[7] F. Peng, D. Schuurmans, S. Wang, and V. Keselj, "Language independent authorship attribution using character level language models," in *Proc. 10th Conf. Eur. Chapter Assoc. Comput. Linguistics-Volume*, 2003, pp. 267–274.
[8] Y. Zhao and J. Zobel, "Searching with style: Authorship attribution in classic literature," in *Proc. 13th Australas. Conf. Comput. Sci. (ACSC)*, Ballarat, VIC, Australia, Jan./Feb. 2007, pp. 59–68.
[9] F. Sebastiani, "Classification of text, automatic," *Encyclopedia Lang. Linguistics*, vol. 14, pp. 457–462, Jan. 2006.
[10] A. Rocha *et al.*, "Authorship attribution for social media forensics," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 5–33, Jan. 2017.
[11] Y. Seroussi, I. Zukerman, and F. Bohnert, "Authorship attribution with topic models," *Comput. Linguistics*, vol. 40, no. 2, pp. 269–310, 2014.
[12] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, 2011.
[13] C. de Roc Boronat and L. Wanner, "On the relevance of syntactic and discourse features for author profiling and identification," in *Proc. EACL*, 2017, p. 681.
[14] M. Eder, "Mind your corpus: Systematic errors in authorship attribution," *Literary Linguistic Comput.*, vol. 28, no. 4, pp. 603–614, 2013.
[15] F. Jannidis, S. Pielström, C. Schöch, and T. Vitt, "Improving burrows' delta—An empirical evaluation of text distance measures," in *Proc. Digit. Humanities Conf.*, 2015, pp. 70–88.
[16] G. Ríos-Toledo, G. Sidorov, N. A. Castro-Sánchez, A. Nava-Zea, and L. Chanona-Hernández, "Relevance of named entities in authorship attribution," in *Proc. Mexican Int. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2016, pp. 3–15.
[17] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, no. 2, p. 7, 2008.
[18] J. Grieve, "Quantitative authorship attribution: An evaluation of techniques," *Literary Linguistic Comput.*, vol. 22, no. 3, pp. 251–270, 2007.
[19] K. Luyckx and W. Daelemans, "The effect of author set size and data size in authorship attribution," *Literary Linguistic Comput.*, vol. 26, no. 1, pp. 35–55, 2011.
[20] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009.
[21] V. Q. Marinho, G. Hirst, and D. R. Amancio, "Authorship attribution via network motifs identification," in *Proc. 5th Brazilian Conf. Intell. Syst.*, 2016, pp. 355–360.
[22] M. Ebrahimpour, T. J. Putniš, M. J. Berryman, A. Allison, B. W.-H. Ng, and D. Abbott, "Automated authorship attribution using advanced signal classification techniques," *PLoS one*, vol. 8, no. 2, p. e54998, 2013.
[23] C. C. Holmes and N. M. Adams, "A probabilistic nearest neighbour method for statistical pattern recognition," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 64, no. 2, pp. 295–306, 2002.
[24] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, 1997.
[25] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, pp. 1746–1751, Sep. 2014.
[26] R. Sarwar, C. Yu, S. Nutanong, N. Urailertprasert, N. Vannaboot, and T. Rakthanmanon, "A scalable framework for stylometric analysis of multi-author documents," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Gold Coast, QLD, Australia, May 2018, pp. 813–829.
[27] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *CoRR*, vol. abs/1509.01626, pp. 649–657, Sep. 2015.
[28] X. Zhang and Y. LeCun, "Text understanding from scratch," *CoRR*, vol. abs/1502.01710, pp. 1–10, Feb. 2015.
[29] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," *Digit. Invest.*, vol. 7, nos. 1–2, pp. 56–64, 2010.

[30] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Automatic text categorization in terms of genre and author," *Comput. linguistics*, vol. 26, no. 4, pp. 471–495, 2000.
[31] F. J. Tweedie and R. H. Baayen, "How variable may a constant be? Measures of lexical richness in perspective," *Comput. Humanities*, vol. 32, no. 5, pp. 323–352, 1998.
[32] T. C. Hoad and J. Zobel, "Methods for identifying versioned and plagiarized documents," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 54, no. 3, pp. 203–215, 2003.
[33] N. Pradhan, M. Gyanchandani, and R. Wadhvani, "A review on text similarity technique used in ir and its application," *Int. J. Comput. Appl.*, vol. 120, no. 9, pp. 29–34, 2015.
[34] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[35] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. 13th Annu. ACM Symp. Theory Comput.*, 1998, pp. 604–613.
[36] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 13th Annu. Symp. Comput. Geometry*, 2004, pp. 253–262.
[37] J. Gan, J. Feng, Q. Fang, and W. Ng, "Locality-sensitive hashing scheme based on dynamic collision counting," in *Proc. SIGMOD*, 2012, pp. 541–552.
[38] R. Lipikorn, A. Shimizu, and H. Kobatake, "A modified Hausdorff distance for object matching," *Pattern Recognit.*, vol. 1, pp. 566–568, Oct. 1994.
[39] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
[40] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.
[41] C. Yu, S. Nutanong, H. Li, C. Wang, and X. Yuan, "A generic method for accelerating LSH-based similarity join processing," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 712–726, Apr. 2017.
[42] S. Nutanong, E. H. Jacox, and H. Samet, "An incremental Hausdorff distance calculation algorithm," *Proc. VLDB Endowment*, vol. 4, no. 8, pp. 506–517, 2011.
[43] G. Hjaltason and H. Samet, "Distance browsing in spatial databases," *ACM Trans. Database Syst.*, vol. 24, no. 2, pp. 265–318, 1999.
[44] S. D. Bay, "Nearest neighbor classification from multiple feature subsets," *Intell. Data Anal.*, vol. 3, no. 3, pp. 191–209, 1999.
[45] C. Mao, B. Hu, P. Moore, Y. Su, and M. Wang, "Nearest neighbor method based on local distribution for classification," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, 2015, pp. 239–250.
[46] J.-P. Posadas-Duran, G. Sidorov, and I. Batyrshin, "Complete syntactic N-grams as style markers for authorship attribution," in *Proc. Mexican Int. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2014, pp. 9–17.
[47] G. Hirst and O. Feiguina, "Bigrams of syntactic labels for authorship discrimination of short texts," *Literary Linguistic Comput.*, vol. 22, no. 4, pp. 405–417, 2007.
[48] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Appl. Intell.*, vol. 19, nos. 1–2, pp. 109–123, 2003.
[49] P. J. Kumar, G. S. Reddy, and T. R. Reddy, "Document weighted approach for authorship attribution," *Int. J. Comput. Intell. Res.*, vol. 13, no. 7, pp. 1653–1661, 2017.
[50] A. S. Altheneyan and M. E. B. Menai, "Naïve Bayes classifiers for authorship attribution of Arabic texts," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 26, no. 4, pp. 473–484, 2014.
[51] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. AAAI Workshop Learn. Text Categorization*, vol. 752, no. 1, pp. 41–48, 1998.
[52] P. Shrestha, S. Sierra, F. Gonzalez, M. Montes, P. Rosso, and T. Solorio, "Convolutional neural networks for authorship attribution of short texts," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 669–674.
[53] D. Rhodes. *Author Attribution With CNNS*. Accessed: Aug. 22, 2016. [Online]. Available: https://www.semanticscholar.org/paper/Author-Attribution-with-Cnn-s-Rhodes/0a904f9d6b47dfc574f681f4d3b41bd840-871b6f/pdf
[54] Z. Yang and B. D. Davison, "Writing with style: Venue classification," in *Proc. 11th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Boca Raton, FL, USA, vol. 1, Dec. 2012, pp. 250–255.
[55] A. Sboev, T. Litvinova, D. Gudovskikh, R. Rybka, and I. Moloshnikov, "Machine learning models of text categorization by author gender using topic-independent features," *Procedia Comput. Sci.*, vol. 101, pp. 135–142, Oct. 2016.
[56] J. Zhu, H. Wang, and X. Zhang, "Discrimination-based feature selection for multinomial Naïve Bayes text classification," in *Proc. Int. Conf. Comput. Process. Oriental Lang.*, 2006, pp. 149–156.
[57] S. Vered. *Representing Words*. Accessed: Jan. 1, 2018. [Online]. Available: http://veredshwartz.blogspot.in/2016/01/representing-words.html
[58] J. Pennington, R. Socher, and C. G. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. EMNLP*, 2014, pp. 1532–1543.

RAHEEM SARWAR received the M.S. degree in computer science from Information Technology University, Pakistan, and the Ph.D. degree in computer science from the City University of Hong Kong. He is currently a Post-Doctoral Fellow with the School of Information Science & Technology, Vidyasirimedhi Institute of Science and Technology, Thailand. His research interests include stylometry, query optimization, and large-scale machine learning.

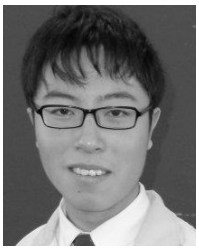
CHENYUN YU received the Ph.D. degree in computer science from the City University of Hong Kong. She is currently a Post-Doctoral Fellow with the Department of Computer Science, National University of Singapore. Her research interests include data processing, query optimization, and large-scale machine learning.

NINAD TUNGARE received the B.S. degree in computer science from the City University of Hong Kong. He is currently pursuing the master's degree in computer science with Boston University. His research interests include stylometry, deep learning, and text classification.

KANATIP CHITAVISUTTHIVONG received the B.S. degree in computer engineering from Kasetsart University, Thailand. He is currently pursuing the Ph.D. degree in information science and technology with the Vidyasirimedhi Institute of Science and Technology, Thailand. His research interests include stylometry, deep learning, and text classification.

SUKRIT SRIRATANAWILAI is currently pursuing the bachelor's degree with the Department of Computer Engineering, Kasetsart University, Thailand. His research interests include stylometry, deep learning, and text classification.

YAOHAI XU is currently pursuing the bachelor's degree with the Department of Computer Science, City University of Hong Kong. His research interests include stylometry, deep learning, and text classification.



DICKSON CHOW is currently pursuing the M.S. degree with the Department of Computer Science, City University of Hong Kong. His research interests include stylometry, deep learning, and text classification.

THANAWIN RAKTHANMANON received the Ph.D. degree in computer science from the University of California, USA. He is currently an Assistant Professor with the Department of Computer Engineering, Kasetsart University, Thailand. His research interests include scientific data management, data-intensive computing, spatial-temporal query processing, and large-scale machine learning. He has authored over 40 journals and conference papers. He has received the 2012 SIGKDD Best Paper Award and the Innovative Award in computer science applications from CHE, Thailand.

SARANA NUTANONG received the Ph.D. degree from the University of Melbourne. He was an Assistant Professor with the City University of Hong Kong. He is currently an Associate Professor with the School of Information Science & Technology, Vidyasirimedhi Institute of Science and Technology, Thailand. His research interests include scientific data management, data-intensive computing, spatial-temporal query processing, and large-scale machine learning. He has authored over 40 journal and conference publications.