IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 March 2024, accepted 17 May 2024, date of publication 30 May 2024, date of current version 6 June 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3407673*

RESEARCH ARTICLE

# Towards Performance Improvement of Authorship Attribution

**AMAR SULJIC AND MD SHAFAEAT HOSSAIN, (Senior Member, IEEE)**
Computer Science Department, Southern Connecticut State University, New Haven, CT 06515, USA
Corresponding author: Md Shafaeat Hossain (hossainm3@southernct.edu)

This work was supported in part by the Connecticut State University (CSU) Research Grant.

**ABSTRACT** An accurate authorship attribution model can play a vital role in security domain by detecting fraudulent texts and combating plagiarism, online piracy, and cyber attacks. In this paper, we work on improving the performance of authorship attribution. To this end, we focus on generating effective samples and features towards creating an authorship attribution model. We did our experiments using a convolutional neural network (CNN). Two key findings from our experiments are as follows: first, our results consistently show that fusing n-grams and stylometric features yields a better performance than independently using each type of features. Notably, with fused features, we achieved an accuracy of 97.03%, a precision of 97.58%, and a recall of 97.03%. Second key finding is—when a sliding window is used in generating training samples, it is possible to improve performance by increasing the amount of overlap between samples, which can be achieved by decreasing the step length of the window. Our study shows that there is a linear relationship between performance metrics and the percent of overlap between training samples. Across three different types of features (n-grams, stylometric, and fused), the worst performance in our experiments was obtained when there was no overlap in the training samples. Inversely, the best performance was achieved when there was a 95% or a 99% overlap in the sliding windows.

**INDEX TERMS** Authorship attribution, fraudulent text, fusion, plagiarism, n-grams, stylometric features.

## I. INTRODUCTION

Every author writes in a way that is unique and distinguishable to them. Thus, by quantifying the patterns and trends found in text, we are able to map a piece of writing to its original author. Authorship attribution has many applications ranging from academia to the security domain, where it has been used to identify plagiarisms, malicious emails, and potential scams (see [1], [2], [3], [4], [5], [6], [7], [8]). Furthermore, it is possible to map source codes back to its original composer, regardless of whether sections of codes were worked on by multiple authors (see [9], [10], [11], [12]). As technology and time progress, so do the applications of authorship attribution.

In this paper, our primary goal is to improve the performance of authorship attribution models. To achieve this goal, we focus on two crucial steps of creating a machine learning model for authorship attribution. Specifically, we work on improving the feature generation and sample generation steps. Below, we introduce the issues in these steps and our approaches to resolve them.

### A. FEATURE GENERATION

Two of the most typical features used for authorship attribution are n-grams and stylometric features [13]. An n-gram is a substring of length n words or characters derived from text [14]. On the other hand, stylometric features are derived from the study of stylometry which is defined as the analysis of features that can be statistically quantified to represent a writing style [13]. Previously, it has been demonstrated that character n-grams are the most successful features in authorship attribution [15], [16], [17], [18]. But, there are also studies which reported remarkable performance using stylometric features [19]. We find that it is necessary to have a direct comparative analysis between character n-grams and stylometric features to determine whether one

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

feature set is stronger than the other for authorship attribution. Additionally, it is imperative to study the effectiveness of fusing n-grams and stylometric features in improving the performance of authorship attribution.

Studies in the past have explored fusion of multiple feature sets [20], [21], [22], [23], [24]. However, these studies fused only stylistic, lexical, perplexity, syntactic, structural, content-specific, or idiosyncratic features which actually fall under the umbrella of stylometric features. That is, there is no study yet which has comprehensively explored the fusion of n-grams and stylometric features. In this study, we close this research gap. We fuse stylometric features with character n-grams alongside implementing character n-grams and stylometric features independently. We test these three different feature sets (stylometric, character n-gram, and fusion) on the same data sets and report respective accuracy, precision, and recall. As a result, we are able to determine which features should be preferred and whether or not they should be implemented independently or fused together. This information is vital in establishing best practices for authorship attribution going forward.

### B. SAMPLE GENERATION

Machine learning models are built based on sample data, sometimes referred to as ''training data''. One approach of generating samples in authorship attribution is to use a sampling window which generates blocks of uniform amounts of characters or words from the text [19]. The sampling window can be used during random sampling or sequential sampling. In the case of sequential sampling, the sampling window is referred to as a sliding window due to the fact that it slides across the text. It has been used in numerous authorship attribution studies [25], [26], [27]. The number of characters or words that the window is shifted after each iteration is known as the window step or step length [27]. Therefore, a window size of 1500 words and a step length of 1500 words means that the window will pass through the text with no overlap between samples. Interestingly, it is possible to make the step length smaller than the window size, in effect creating an overlap between samples and as a result generating more samples. This approach, however, has yet to be studied with many questions left unanswered such as whether or not increasing the percentage of overlap between samples will lead to improved model performance.

Our study provides a deep look into the relationship between the overlap percentage used in a sliding window during sample generation and the respective accuracy, precision, and recall. Furthermore, we explore this problem using n-grams, stylometric features, and a concatenation of stylometric features and n-grams.

### C. SUMMARY OF OUR CONTRIBUTION AND NOVELTY

We work on generating enhanced training samples and identifying the best features for developing authorship attribution models. The advancement of authorship attribution performance helps different sectors such as academia, cybersecurity, and digital forensics. There is still uncertainty in the field as to the best features to use and a continual demand for more data to train models on. By validating a means of generating more training samples and providing analysis between the best features for authorship attribution, we improve the ability to map a piece of writing back to its original author. Below, we list our novelty:

- We provide empirical analysis between n-grams, stylometric features, and a fusion of both feature types. In addition to performance metrics, this analysis includes the amount of time it took to generate samples, preprocess samples, extract features from samples, fit models, and the amount of time it took to classify new samples. Each part of the classification process was carefully timed to provide an extensive resource for future work within authorship attribution.
- We show that fusing n-grams and stylometric features yields higher accuracy, precision, and recall than using n-grams or stylometric features independently. This result is significant because it encourages future work within authorship attribution to consider fusing n-grams and stylometric features as opposed to using one over the other to achieve a higher performance. Additionally, through this analysis we show that an optimal model that considers both performance and time is heavily dependent on the features chosen. That is to say, n-grams and stylometric features should be tested both independently and fused together on the same data before an optimal model is decided on.
- Our study used varying amounts of overlap percentages with a fixed window size, which allowed us to isolate our focus on the impact that the overlap percentage has on model performance. As a result, we show that the performance of a model can be significantly improved by increasing the amount of overlap found between samples in a sliding window.

For the classification method, we chose to use a Convolutional Neural Network (CNN). Several recent studies have demonstrated that Deep Learning classifiers such as the CNN are highly effective in doing authorship attribution tasks [28], [29], [30], outperforming traditional machine learning methods [19], [20], [31]. Hence, we find it imperative to perform the analysis using the CNN.

For creating the training and testing datasets for the CNN model, we chose to sample literature books to explicitly explore solutions for the plagiarism issue in academia. Regardless of the advancements of technology, there is always a need for anti-plagiarism capabilities in academia. The long format of the text allows us to sample our data the same way sampling could be done for college essays, graduate thesis, and peer-reviewed papers.

The rest of the paper is organized as follows: Section II discusses related works on the topic of authorship attribution. Section III introduces our data and discusses our use of the sliding window. In Section IV, we discuss and explain the

features used. Section V explores the architecture chosen for the CNN. The experiment settings are discussed in Section VI. In Section VII, we report and discuss the results of our experiments and finally in Section VIII, we state the conclusion of our work.

## II. RELATED WORKS

### A. STUDIES ON STYLOMETRIC AND N-GRAM FEATURES

Stylometric and n-gram features have been extensively used in authorship attribution. Reference [20] reported an accuracy of 97% using a novel plagiarism detection approach. In this study, the statistical properties of the most frequent words were analyzed and stylometric features were generated. Another example is [19] who used 13 stylometric features on public domain literature and reported 98.10% accuracy, 98.05% precision, and 98.04% recall. Samples were generated through random sampling using a sampling window of 1500 words. So, random points within the text were selected and blocks of 1500 words were extracted. Since random sampling was used, we can deduce that some overlap can be found between samples. In-fact, the best accuracy, precision, and recall were reported by the largest data set of 500 samples per author. Therefore, we can assume that the larger data sets performed better due to the fact that there was inherently more overlap found between samples. One limitation to this study is that random sampling means there can be some overlap between training and test samples. Reference [25] developed a stylometric representation learning approach for authorship analysis and demonstrated its effectiveness in three different problems, namely authorship identification (or attribution), verification, and characterization. Reference [32] shows how n-grams can be used to evaluate the cosine, dice, extended Jaccard's, and overlap similarities between works. It was found that the overlap similarity was the most effective because it consistently detected texts that had remarkable amounts of identical passages, which is plagiarism.

There are some studies which attempted to distinguish which features perform better between stylometric and n-grams. Reference [33] compared the effectiveness of stylometric features and n-grams and concluded that n-grams reported higher accuracy than stylometric features across three algorithms. A limitation of this study, however, is their implementation of lexical features expands beyond just n-grams. For instance, the number of nouns, verbs, and adjectives were included under the lexical category. We believe that there should be a clear distinction between n-grams and features which quantify text. Thus, there was not a direct comparison between n-grams and stylometric features. A recent study done by [34] explored the differences between n-grams and stylometric features in a more comprehensible way. The results of this study showed that idiosyncratic features such as the number of misspelled words, abbreviations, and slang were the best features for authorship classification of digital text reporting an accuracy of 98.5%, outperforming n-grams by close to 2%. Another significant contribution of this study is that character level and word level n-grams were both tested, with various values for n. In the end, it was found that character level n-grams performed better than word level n-grams. Additionally, the study noted that fusing n-grams and stylometric features should be considered for future work to combat some of the issues discovered during the classification process. Reference [35] did an in-depth analysis between 162 stylometric features and various levels of n-grams using a decision tree as the classifier. In their study, it was concluded that n-grams as a whole were far better at correctly mapping a text to its original author than stylometric features were. Specifically, the best accuracy for n-grams was generated using 3-grams reporting 52.82%, while the best accuracy for the 162 stylometric features was reported to be 35.26%.

A number of studies attempted to improve the performance of authorship attribution by fusing information from multiple modalities or feature sets. Reference [36] compared a number of classifiers and feature-level fusions and reported that the best accuracy (96.3%) was achieved by fusing part of speech (POS), character, and word level features. Furthermore, [24] demonstrated that generating meta features from the original feature set provided better accuracy than using the original feature set alone. Reference [23] also demonstrated that an extended feature set outperformed sets of features that came from one modality.

The primary difference between above fusion-based studies and ours is that we isolate stylometric features from n-grams. The feature groups {Stylistic, Lexical, Perplexity, and Syntactic} used in [24], {Lexical, Syntactic, Structural, Content-specific} used in [20], [21], and [22], and {Idiosyncratic} used in [23] all fall under the umbrella of stylometric features in our study. Reference [35] performed a more similar study to ours in that they compared different levels of n-grams with a feature set that included stylometric features and n-grams. Using decision trees, they found that on one test set, the fused feature set performed the best (40% accuracy, precision, recall), while on the other test set, the fused feature set (49.95% accuracy and recall, 50% precision) failed to outperform n-grams (50.21% accuracy, 52.31% precision, 52.07% recall). Though, given the minute difference in performance, it was concluded that fusing stylometric features and n-grams did not improve the performance. The conclusion of our work is different.

### B. STUDIES ON SAMPLE GENERATION

Finding ways to generate samples for authorship attribution has scarcely been studied. There are some studies which have used the sliding window technique for sample generation [26], [27]. Reference [26] calculated the dissimilarity between two adjacent sliding windows to distinguish whether the selected texts belong to the same author. Additionally, [27] compared the performance of using a sliding window for authorship attribution against not using one. References [19] and [37] used the random sampling technique for sample

**TABLE 1.** State-of-the-art authorship attribution schemes. SVM: Support Vector Machine, WP: Writeprints, KNN: K-Nearest Neighbor, ANN: Artificial Neural Network, NB: Naive Bayes, MLP: Multi Layer Perception: RF: Random Forest, LSTM: Long Short Term Memory, GRU: Gated Recurrent Unit,, DT: Decision Tree, CD: Cosine Distance, MD: Manhattan Distance, ED: Euclidean Distance, TFIDF: Term Frequency-Inverse Document Frequency, Acc.: Accuracy, Prec.: Precision, Rec.: Recall.

| Author /year | Data | Feature | Classi -fier | Fusion of stylometric & n-gram | Overlap sampling | Timing info | Performance |
|---|---|---|---|---|---|---|---|
| Zheng et al. (2006) [21] | Online message boards | Stylometric | SVM | No | No | No | Acc. 97.69% (English dataset), Acc. 88.33% (Chinese dataset) |
| Li et al. (2006) [22] | Online message boards | Stylometric | SVM | No | No | No | Acc. 99.01% (English dataset), Acc. 96.56% (Chinese dataset) |
| Abbasi & Chen (2008) [23] | Emails, messaging, comments, code | Stylometric, n-grams | WP, SVM | No | No | No | Acc. 100% (25 authors), Acc. 97.96% (50 authors), Acc. 94.59% (100 authors) |
| Solorio et al. (2011) [24] | Online forum posts | Stylometric, meta | SVM | No | No | No | Acc. 76.17% (5 Authors), Acc. 77.38% (10 authors), Acc. 71.42% (20 authors), Acc. 63.79% (50 authors), 62.10% (100 authors) |
| Ramnial et al. (2016) [2] | PhD theses | Stylometric | SVM, KNN | No | No | No | Acc. Prec. Rec. 94% (1,000 words), Acc. Prec. Rec. 98% (5,000 words), Acc. Prec. Rec. 98% (10,000 words) |
| Boran et al. (2016) [37] | Literature books | Stylometric | SVM, ANN, NB | No | No | No | Acc. 83% (small dataset), Acc. 89% (medium dataset), Acc. 90% (large dataset) |
| Alsallal et al. (2017) [20] | Literature books | Stylometric | SVM, MLP, RF | No | No | No | Acc. 97% |
| Sari et al. (2017) [30] | Judge rulings, news stories, movie reviews | N-grams | SVM, CNN | No | No | No | Acc. 92% (Judgement), Acc. 77% (CCAT10), Acc. 73% (CCAT50), Acc. 95% (IMDb62) |
| Shrestha et al. (2017) [39] | Tweets | N-grams | CNN | No | No | No | Acc. 76% |
| Gupta et al. (2019) [29] | Online news sources | N-grams | LSTM, GRU | No | No | No | Acc. 78% (C50), Acc. 97% (BBC) |
| Abuhamad et al. (2019) [28] | Tweets | TFIDF | CNN | No | No | No | Acc. 96% (C++), Acc. 96% (Java), Acc. 95% (Python) |
| Fourkioti et al. (2019) [36] | Tweets, online movie reviews | Stylometric, n-grams | SVM, KNN, RF, NB | No | No | No | Acc. 96% (movie reviews), Acc. 54% (tweets) |
| Shang et al. (2020) [35] | Chinese literature books | Stylometric, n-grams | DT | Yes | No | No | Acc. 53%, Prec. 54%, Rec. 53% |
| Belvisi et al. (2020) [34] | Tweets | Stylometric, n-grams | CD, MD, ED | No | No | No | Acc. 98.5% |
| Boran et al. (2020) [19] | Literature books | Stylometric | CNN | No | No | No | Acc. 98.10%, Prec: 98.05%, Rec: 98.04% |

generation and achieved promising performance because of inherent overlap between samples. Reference [31] used a thesaurus to change some words and in effect create new samples. The study concluded that text should not be altered because the samples will lose their inherit qualities and meanings. Another study done by [38] showed that text could be altered by replacing words with those that had the highest cosine similarity. In that study, it was found that better classification performance was obtained from the altered text than from the original. Our study, however, does not alter the original text in any way. To create additional samples, we decrease the step length of the sliding window. In return, there is a greater amount of overlap found between samples, which leads to additional samples.

### C. STUDIES USING DEEP LEARNING IN AUTHORSHIP ATTRIBUTION

Deep Learning is growing increasingly more popular in authorship attribution given the fact that Deep Learning models are able to outperform traditional machine learning methods [19]. Reference [28] showed how effective CNNs are for authorship attribution, correctly classifying the author of code samples over 90% of the time across three programming languages. Another instance of the utilization of Deep Learning in authorship attribution is [29] where two different Deep Learning models were implemented using n-grams and their performance metrics were compared. The conclusion of the experiments was that the Gated Recurrent Unit Network yielded higher accuracy than the Long Short Term Memory Network across all datasets. Furthermore, [29] proposed a new embedding method and tested it against pre-trained embeddings which proved to perform worse than the proposed embedding method. Lastly, [20] sampled literature books and used a Multilayer Perception Network (MLP) to classify authors. The MLP reported an accuracy of 97%, outperforming the Support Vector Machine and Random Forest classifiers.

### D. HOW DOES OUR WORK ADVANCE THE STATE-OF-THE-ART?

In Table 1, we highlight the key aspects from the state-of-the-art studies in authorship attribution. Below we identify the research gap in the state-of-the-art schemes and clarify how our work closes this gap.

- **Fusion of stylometric and n-gram features**: There are a number of studies that used stylometric and n-gram features in authorship attribution. However, there is a lack of research that demonstrates the effectiveness of fusing stylometric and n-gram features. There are some studies such as [20], [21], [22], [23], and [24] which fused stylistic, lexical, perplexity, syntactic, structural, content-specific, or idiosyncratic features which actually fall under the type of stylometric features. We found only one study ([35]) that attempted to fuse stylometric and n-gram features, however, it could not demonstrate significant performance improvement. We believe it is

possible to improve the performance if stylometric and n-gram features are fused properly and our experimental results reflect that.

- **Varying amounts of overlap percentages in sample generation:** While the sliding window technique has been used for sample generation in several authorship attribution studies [23], [26], [27], there is no study yet that has implemented varying step lengths for a sliding window to explore the effect that overlap percentage has on classification performance. In our study, we close this research gap and demonstrate significant performance improvement by increasing the amount of overlap.
- **Analyzing timing information**: Existing studies only reported how accurate an authorship attribution scheme was. They completely ignored how long it took to preprocess the data, extract features, train the model, and test a new sample. We opine that this timing information is crucial in comparing authorship attribution schemes. For this reason, we have timed all portions of our experiments. When comparing our models, we do not just look at the performance metrics; instead, we provide an overall analysis that considers the timing information also. This analysis will help practitioners make informed decisions on a use case basis.
- **Addressing the plagiarism issue in academia**: A noticeable gap in the related works is that the majority of them focused on shorter form samples such as emails or social media posts. As a result, there are less solutions for plagiarism problems in academia. A study on data collected from over 80,000 students and 12,000 faculty in the United States and Canada shows that 38% of undergraduates have plagiarized a written source, 36% have plagiarized a digital source, 14% have fabricated a bibliography, and 8% have turned in work that was done by another researcher [40]. In our study, we create long samples from literature books, which resemble the samples created from college essays, graduate thesis, and peer-reviewed papers, and thus allow us addressing the plagiarism issue in academia.

## III. DATA

Data was collected from the online repository Project Gutenberg.[1] Project Gutenberg contains public domain literature that can be downloaded in plain text format. In total, we downloaded 50 books, 5 from 10 authors each. Table 2 lists out the 10 authors alongside some additional information such as when they were alive. From each book we manually removed any publishing and licensing agreements on top of the table of contents. Next, the contents of each book were converted to lowercase and the books were cleaned of any foreign or non interpretable characters. We chose to do this before generating samples so that we could avoid generating any samples with large amounts of non interpretable characters which would in essence destroy the sample. Also, the reason that the text was converted to lowercase was to avoid any conflicts during feature extraction.

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

**TABLE 2. The name of the authors sampled, the years they were born and died and finally the number of samples per author in the 0% overlap set.**

| Author | Lived | # of Samples (0% overlap) |
|---|---|---|
| James F. Cooper | (1789-1851) | 440 |
| Charles Dickens | (1812-1870) | 497 |
| Arthur C. Doyle | (1859-1930) | 202 |
| Fyodor Dostoevsky | (1821-1881) | 590 |
| Nathaniel Hawthorne | (1804-1864) | 258 |
| Herman Melville | (1819-1891) | 357 |
| Washington Irving | (1783-1859) | 249 |
| Jack London | (1876-1916) | 326 |
| Jacob Riis | (1849-1914) | 238 |
| Mark Twain | (1835-1910) | 293 |

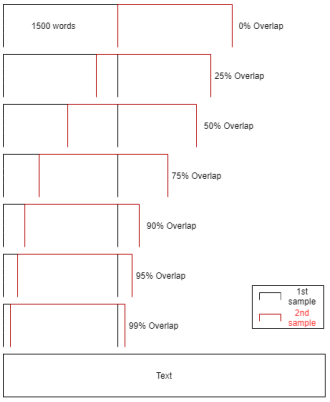



**FIGURE 1. A depiction of how samples were generated for each overlap percentage. The pattern in each overlap set continues until max samples are generated from each book. *Note–figure is not to scale.***

We chose to sequentially generate our samples to avoid any bias that may come from random sampling such as heavily sampling a subsection of a book. The books were sampled using a sliding window with a length of 1500 words. In essence, each sample contains 1500 words. The window length is inspired by a previous study done by [19] who used a window length of the same size on similar data. The window length was chosen because of the variety of content in the books. For instance, if the window size was smaller, then it is possible to only extract passages from the text that are conversations or monologues from a character, which would not be representative of the author's writing style and more so how the character in the book is feeling [19].

To generate various sized data sets we increased the overlap found in the window by decreasing the step length which is the number of words the window slides between samples. For instance, if we have 0% overlap in the sampling window, then our step length would match our window length of 1500 words. But, to generate samples with a 25% overlap, the step length would be smaller, in our case 1125 words. Figure 1 further breaks down how samples were generated using a sliding window and Table 3 shows the various data sets generated.

During the sampling with 0% overlap, every 5th sample was sent to the test set and was then removed from the book

**TABLE 3. A break down of the number of training samples generated per each overlap percentage alongside the corresponding step length of the sliding window used.**

| Data set # | Window overlap | Step length (words) | Number of samples |
|---|---|---|---|
| 1 | 0% | 1500 | 3,450 |
| 2 | 25% | 1125 | 3,697 |
| 3 | 50% | 750 | 5,520 |
| 4 | 75% | 375 | 11,006 |
| 5 | 90% | 150 | 27,446 |
| 6 | 95% | 75 | 54,859 |
| 7 | 99% | 15 | 274,145 |

**TABLE 4. Stylometric features used in [19] as well as in this study.**

| Stylometric Features | |
|---|---|
| Number of Paragraphs | Number of Pronouns |
| Numbers of Adverbs | Number of Alliterations |
| Number of Words Ending in '-ing' | Number of Characters per Sentence |
| Number of Long Words | Short/Long Word Ratio |
| Number of Colons | Number of Semicolons |
| Number of Commas | Number of Hyphens |
| Number of Exclamation Points | |

entirely. By removing the testing samples we are able to increment the overlap within our window without any fear of creating bias between training and validation sets. In the end, 843 samples were generated for testing.

## IV. FEATURES

There are three sets of features used in our experiments, stylometric, n-grams, and a fusion of n-grams and stylometric features. For stylometric features, we follow [19] who reported an excellent performance using the 13 features located in Table 4. A study prior to the one mentioned above also done by [37] performed an in-depth analysis into the effectiveness of many of these features. As a result, the stylometric features chosen are ones that have proven to be effective. For the number of alliterations we decided to count 3 consecutive words starting with the same letter as an alliteration. As far as quantifying the short to long word ratio, we deemed any word greater than 5 characters as a long word as done by [41].

One of the underlying principles that allows for authorship classification to occur successfully is that everyone has a unique writing style that pertains to them. To illustrate this point, we used our largest data set (274,145 samples) to plot the mean of the stylometric features for each respective author, found in Figure 2. The fact that no stylometric feature is plotted as a straight line gives credence to the fact that writing styles are unique and distinguishable when quantified. The plot not only shows that there is no uniform writing style amongst our authors, but also depicts in which ways the writing style of each author differed. Conversely, we can look for less dramatic lines and observe in which ways the writing styles of our authors were similar. As a whole, Figure 2 reinforces the key concepts in authorship attribution that make mapping a piece of text to its original author possible.

The best value to use for n with n-grams has been a frequent focus of studies in the past. For this study, we rely

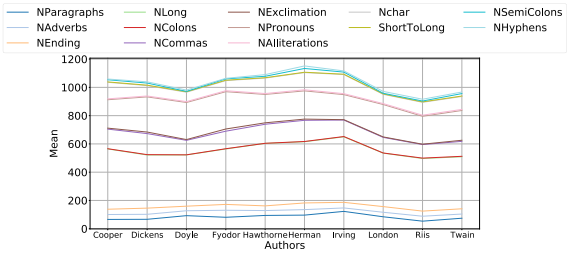



**FIGURE 2. An illustration of the average values per author for each stylometric feature using 274,145 samples. Each feature is represented in Table 4.**

on previous studies that have compared the effectiveness between different levels of n, and between character and word level n-grams [30], [34]. Specifically, we followed [30] where continuous character level n-grams up to four-grams, reported the best results amongst other character and word level n-grams.

A major difference between using stylometric features and n-grams is the amount of preprocessing needed per sample. While stylometric features just require text to be converted to lowercase and stripped of foreign characters, n-grams additionally require for the text to be encoded to the set vocabulary. Our experiment used the same vocabulary of 70 characters as used by [30] and [31].

Finally, for the fused features, we concatenated the stylometric and n-gram features described above. The features were fused prior to being fed to a model so a single sample contains both features. For example, we had 13 stylometric features and 9000 n-gram features, when concatenating both we end up with 9013 features.

## V. CNN ARCHITECTURES

A Convolutional Neural Network (CNN) was chosen for our experiments because recent authorship attribution studies demonstrated high classification performance by using CNNs [30], [31], [39], [42]. Additionally, [19] showed that given larger amounts of samples, the CNN outperformed traditional machine learning classifiers. For our experiments, we used two different CNN architectures: one for stylometric features and another for n-grams and fused features. Table 5 illustrates the difference in structure. The main difference between the two architectures is that we used a 2D CNN for the stylometric feature and a 1D CNN for the n-grams and fused features. We used a 1D CNN for n-gram and fused features to follow the approach of [30] and due to the fact that we needed an embedding layer which is only offered in the 1D implementation. During preliminary experiments, we found that the 2D CNN was performing better with stylometric features than the 1D CNN, therefore, we choose to continue with the 2D implementation for stylometric features.

To determine the best hyper-parameters for the CNN, we implemented GridSearchCV from the Scikit-learn library. GridSearchCV allowed us to implement cross validation which is when the data is partitioned into an equal number of sets and from there one set is isolated for testing and

**TABLE 5.** The architecture of the two CNN variations.

| CNN-Stylometric | |
|---|---|
| Conv2D | filters = 32, kernel_size = (3, 3) |
| MaxPooling2D | pool_size = (2, 2), padding = same |
| Flatten | |
| Dense | L2 regularizer |
| Dropout | |
| Dense | |
| Output | activation = softmax, optimizer = adam<br>loss = categorical_crossentropy |
| **CNN-N-gram/Fusion** | |
| Embedding | initializer = glorot_uniform |
| AveragePooling1D | pool_size = 10 |
| Flatten | |
| Dense | L2 regularizer, activation = relu |
| Dropout | |
| Dense | L2 regularizer, activation = relu |
| Output | activation = softmax, optimizer = adam<br>loss = categorical_crossentropy |

**TABLE 6.** The hyperparameters and values searched for our CNN.

| Parameters | Values Tested |
|---|---|
| Hidden Nodes | (100-500, 200-1000) |
| L2 | .1, .01, .0001 |
| Dropout | 0, .1, .2, .3, .4, .5, .6, 7, .75 |
| Optimizer | adam, sgd |
| Activation Function | ReLU, Tanh |

the rest for training. The test set will change every time, which maximizes the generalization of the model. For our experiments, we chose to separate our data into 3 equal partitions. Those values that we tested can be found in Table 6 and the hyperparameters chosen can be found in Table 7.

## VI. EXPERIMENT SETTINGS

This experiment was completed in Google Colab Pro using the provided GPU. During the initial iterations of our experiments, it was found that we needed to upgrade to Google Colab Pro to get more RAM. The original RAM given was about 12GB which did not allow us to exceed 75% overlap with n-grams and fusion. Once upgraded, we received 25GB of RAM which allowed us to complete experiments using n-grams and fused features up to 95% overlap. Everything was done using Python 3.7.4, and TensorFlow was used to implement the CNN. Early stopping was implemented monitoring the validation loss with a patience of 5 for the CNN. What this means is that if the validation loss did not go down after 5 epochs of training, then the model would stop training. For this reason, the number of epochs needed per experiment varies from 4 to 500. The experiments were timed using the Python time function which allowed us to save the current time as a variable. Prior to executing code, the current time would be saved in a variable T. Once the code finished executing, we would subtract the variable T from the new current time to get the amount of time that has passed.

## VII. RESULTS AND DISCUSSION

### A. PERFORMANCE EVALUATION

The metrics we used to evaluate the performances of our models were the accuracy, precision, and recall. A brief explanation of the metrics is as follows. Accuracy is defined as the number of correct predictions to the number of total predictions. Precision is understood as the correctly identified positive classifications to the total predicted positive classifications. Lastly, recall can best be understood as the amount of times a class was positively classified to the total true positive and false negative classifications. Figure 3 depicts the accuracy, precision, and recall for each feature type per overlap percentage. Our observations from this figure are given below.

*Does increasing percentage of window overlap yield better accuracy, precision, and recall?* Our experiments show that increasing the percentage of overlap does lead to higher accuracy, precision, and recall. Figure 3a shows that there is a linear relationship between the accuracy and overlap percentage for all three feature types. The same linear pattern is found when using n-grams and fused features for both the precision and recall as shown in Figures 3b and 3c. Even though the precision and recall are not entirely linear when using stylometric features, they are still better at 99% overlap than 0%. Thus, we can conclude that the greater the percentage of overlap between samples, the greater the accuracy, precision, and recall.

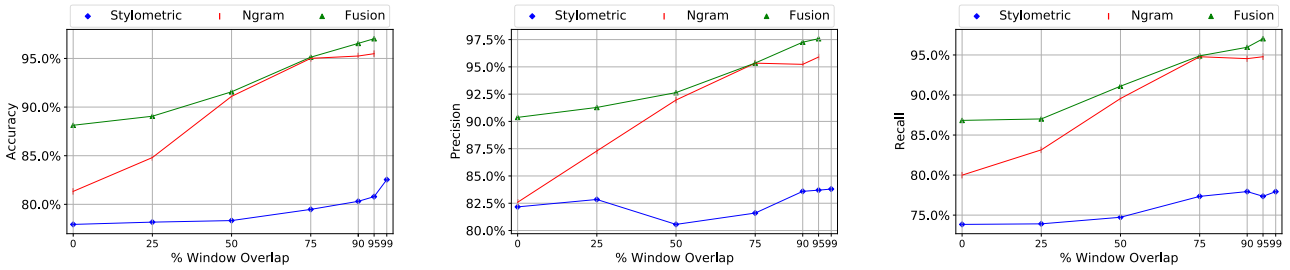
*Why does increasing overlap percentage result in higher performance metrics?* The reason that the percentage of overlap plays such a pivotal role in performance metrics is because the higher the percentage of overlap is the more training samples there are to train the model with. For instance, Table 3 shows that by increasing our overlap from 0% to 99% we can generate 270,695 additional unique samples. The more training samples we provide our model, the more information it has to make a classification. We can assume that this extra information is helpful in making correct classifications given that the extra samples are meaningful.

*Which features performed the best?* Figure 3 shows us that fused features achieved higher accuracy, precision, and recall consistently across all sample sizes. At no point did n-grams or stylometric features report better metrics than fused features. Table 8 contains the best accuracy, precision, and recall reported by each feature type, where under the metric there is the respective overlap percentage which was used to generate the metric. From here, we can observe that the best accuracy (97.03%), precision (97.58%) and recall (97.03%) were reported by fused features. On the other hand, the worst rates were reported by stylometric features, with a difference in accuracy, precision and recall of 14.49%, 13.78% and 19.09% respectively. The difference in performance between fused features and n-grams is not as severe with a maximum difference of 2.25% between their respective recalls. Therefore, we can conclude that the best features to use for authorship attribution are a fusion of stylometric and n-grams.

*Further analysis on the relationship between overlap percentage and the performance metric:* The results of our experiments showed an almost linear relationship between overlap percentage and the performance metrics of a classifier. Regression was used in an effort to further explore

**TABLE 7.** The hyper parameters used for each feature type, per overlap percentage. Act: Activation, Opt: Optimizer.

| Window Overlap | Features | Layer1 Nodes | L2 | Dropout | Layer2 Nodes | Epochs | Act | Opt |
|---|---|---|---|---|---|---|---|---|
| 0% | Stylometric | 220 | 0.0001 | 0.3 | 400 | 500 | relu | adam |
| | N-grams | 100 | 0 | 0.5 | 200 | 10 | relu | adam |
| | Fusion | 100 | 0 | 0.5 | 200 | 12 | relu | adam |
| 25% | Stylometric | 220 | 0.0001 | 0.3 | 440 | 500 | relu | adam |
| | N-grams | 120 | 0 | 0.7 | 220 | 9 | relu | adam |
| | Fusion | 120 | 0 | 0.7 | 220 | 10 | relu | adam |
| 50% | Stylometric | 240 | 0.0001 | 0.2 | 440 | 500 | relu | adam |
| | N-grams | 140 | 0 | 0.5 | 200 | 12 | relu | adam |
| | Fusion | 140 | 0 | 0.7 | 200 | 11 | relu | adam |
| 75% | Stylometric | 200 | 0.0001 | 0.1 | 400 | 500 | relu | adam |
| | N-grams | 140 | 0 | 0.5 | 220 | 10 | relu | adam |
| | Fusion | 140 | 0 | 0.5 | 220 | 10 | relu | adam |
| 90% | Stylometric | 200 | 0.0001 | 0.1 | 800 | 500 | relu | adam |
| | N-grams | 140 | 0 | 0.5 | 220 | 7 | relu | adam |
| | Fusion | 140 | 0 | 0.5 | 220 | 9 | relu | adam |
| 95% | Stylometric | 200 | 0.0001 | 0.1 | 800 | 500 | relu | adam |
| | N-grams | 140 | 0 | 0.5 | 240 | 4 | relu | adam |
| | Fusion | 140 | 0 | 0.5 | 240 | 6 | relu | adam |
| 99% | Stylometric | 400 | 0.0001 | 0.3 | 800 | 500 | relu | adam |



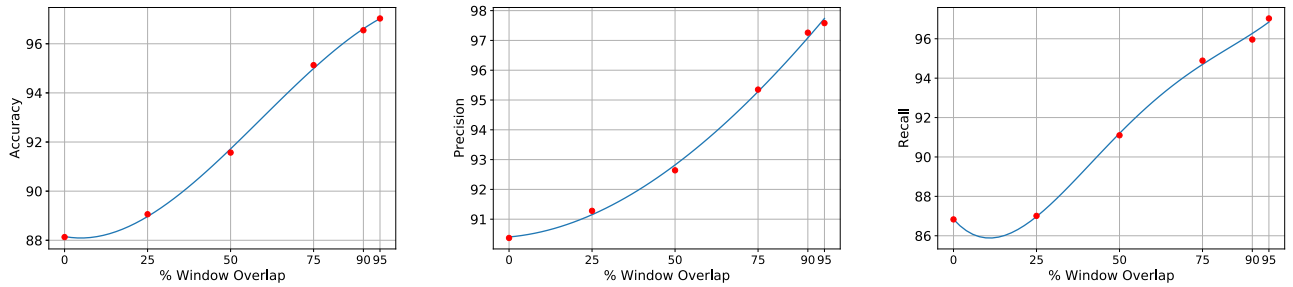

(a) Accuracy (b) Precision (c) Recall

**FIGURE 3.** The results of the experiments using the different feature sets and overlap percentages.

**TABLE 8.** The best performance results achieved per feature type. Additionally, under the metric, the overlap percentage (s) at which the metric was achieved is included.

| Features | Accuracy | Precision | Recall |
|---|---|---|---|
| Stylometric | 82.54% (99%) | 83.80% (99%) | 77.94% (99%) |
| N-gram | 95.49% (95%) | 95.90% (95%) | 94.78% (95%, 75%) |
| Fusion | 97.03% (95%) | 97.58% (95%) | 97.03% (95%) |

this relationship. Initially we used linear regression but soon moved to polynomial regression where we saw satisfactory fits. Figure 4 shows the results of the fits on accuracy, precision, and recall using fused features. Fused features are highlighted here because fused features achieved the best performance metrics in our experiments. Thus, there is some credence that the performance of a model can be predicted given the percentage of overlap used in the sliding window. Beyond that, one could work the equations backwards and find what overlap percentage they should use to achieve their desired metrics. In essence, given an equation that corresponds to the model's performance allows us to optimize our models without writing any code. There are many reasons why it would be helpful to have an idea of the performance of the model even prior to implementation. The main reason is that it saves time in trying to find the optimal step length (or window overlap percentage). The different overlap percentages can be plugged into regression equations instead of doing multiple implementations with various overlap percentages in a trial and error format. Additionally, as mentioned before the equations can be optimized for a desired metric. That is to say, we can work backwards and see what overlap percentage we should use within our sampling window to achieve a desired metric. This type of application becomes extremely useful when dealing with extreme amounts of data because sometimes it is not convenient or possible to sample with 99% overlap. Or, like in our case, there is not enough RAM to sample with 99% overlap. Therefore, one can find the minimum overlap needed to achieve the desired metric, which in turn minimizes the time spent in sampling but also maximizes model performance. To select the lines of best fit we used cross validation (CV = 3) and chose the fits which minimized the root mean squared error (RMSE). The Pearson correlation coefficient was 0.986 for the accuracy, 0.978 for the precision, and finally 0.978 for the recall. Therefore, we can see that there is a strong correlation between overlap percentage and accuracy, precision, and recall.




(a) $Y = -2.51\text{e-}02x^3 + 2.689\text{e-}03x^2 - 1.517\text{e-}05x + 88.15$ (b) $Y = 9.754\text{e-}03x^3 + 8.403\text{e-}03x^2 - 1.373\text{e-}06x + 90.40$ (c) $Y = -1.892\text{e-}01x^4 + 1.069\text{e-}02x^3 - 1.292\text{e-}06x^2 + 5.193\text{e-}07x + 86.84$

**FIGURE 4.** The best fit curves for overlap percentages and performance metrics when experiments were performed using fused features. The sub-figure caption shows the respective regression equation for the curve.

**TABLE 9.** The amount of seconds needed for sample generation, data preprocessing and feature extraction per overlap percentage for stylometric, n-grams, and fused features.

| Window Overlap | Sample Generation | Stylometric | | | | N-gram | | | | Fusion | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Data Preprocessing | | Feature Extraction | | Data Preprocessing | | Feature Extraction | | Data Preprocessing | | Feature Extraction | |
| | | All Samples | Per Sample | All Samples | Per Sample | All Samples | Per Sample | All Samples | Per Sample | All Samples | Per Sample | All Samples | Per Sample |
| 0% | 6.839 | 0.41 | 0.0001 | 314.51 | 0.091 | 519.24 | 0.151 | 73.87 | 0.021 | 519.65 | 0.150 | 388.38 | 0.113 |
| 25% | 6.848 | 0.45 | 0.0001 | 417.80 | 0.113 | 682.78 | 0.185 | 114.00 | 0.031 | 683.23 | 0.185 | 531.80 | 0.144 |
| 50% | 8.945 | 0.55 | 0.0001 | 629.15 | 0.114 | 964.55 | 0.175 | 138.15 | 0.025 | 965.10 | 0.175 | 767.30 | 0.139 |
| 75% | 15.062 | 2.66 | 0.0002 | 1,222.41 | 0.111 | 1,618.20 | 0.147 | 273.29 | 0.025 | 1,620.86 | 0.147 | 1,495.70 | 0.136 |
| 90% | 40.390 | 5.88 | 0.0002 | 3,196.45 | 0.116 | 3,317.96 | 0.121 | 515.37 | 0.019 | 3,323.84 | 0.121 | 3,711.82 | 0.135 |
| 95% | 81.342 | 7.27 | 0.0001 | 6,629.35 | 0.121 | 6,744.28 | 0.123 | 1,043.61 | 0.019 | 6,751.56 | 0.123 | 7,672.96 | 0.140 |
| 99% | 335.726 | 29.22 | 0.0001 | 31,094.13 | 0.113 | - | - | - | - | - | - | - | - |

### B. TIMING EVALUATION

In addition to the analysis of the performance metrics of the models, analysis was done on the amount of time needed per feature set to complete certain stages of the model deployment process. This process starts with generating samples and ends with evaluating our model on the test set.

*What features were preprocessed the quickest?* Table 9 shows us that per sample, stylometric features were preprocessed the quickest across all overlap percentages. On the other hand, fusion features were preprocessed the slowest. Between stylometric features and n-grams, the difference of efficiency is immediately noticeable starting with 3,450 samples (0% overlap) and at 54,859 samples (95% overlap) the difference grows to 6,737 seconds or just under 2 hours. Looking from the per sample perspective, at 95% overlap, stylometric features were preprocessed over 1000 faster than n-grams. In the end, we can conclude that stylometric features are preprocessed quickest among n-grams and fused features.

*What features were extracted the quickest?* Table 9 gives significant insight between the stylometric features and n-grams showing that n-grams are extracted from text more efficiently than stylometric features are. The contrast of efficiency becomes more dramatic as the sample size grows. For instance, with a window overlap of 95% (54,859 samples), n-grams were extracted 6 times faster per sample than the stylometric features were.

*What features fit a model most efficiently?* Table 10 shows only the fit time for each feature set. We can see that the time it took to fit a model using stylometric features not only scales well with sample size, but also is faster than n-grams and fused features. Table 10 also shows that the amount of time it takes to fit a model using n-grams is not always linear with the amount of samples a model has. This is because a model may reach the maximum performance metrics quicker with more samples.

**TABLE 10.** The amount of seconds it took to fit each feature set per overlap percentage. This only includes fit time.

| Overlap | Stylometric | N-gram | Fusion |
|---|---|---|---|
| 0% | 22.47 | 341.91 | 358.62 |
| 25% | 22.64 | 414.81 | 349.31 |
| 50% | 25.81 | 741.74 | 465.79 |
| 75% | 27.48 | 537.47 | 523.95 |
| 90% | 39.46 | 2005.73 | 1148.27 |
| 95% | 43.22 | 2086.24 | 2402.98 |
| 99% | 164.65 | - | - |

*What features make decisions on new data most efficiently?* Table 11 shows the number of seconds that it took to make a prediction on a new sample. However, this only includes the time it took for the model to make a decision. The true testing time encompasses the time it took to generate samples, preprocess samples, extract features from samples and finally classify samples. Table 12 includes sample generation, preprocessing, feature extraction and classification time and shows us the total number of seconds it took to classify a new sample for each feature set at each overlap percentage. We observe that stylometric features

**TABLE 11.** The number of seconds it took to make a prediction per sample, for each overlap percentage. This only includes prediction time and not sample generation, preprocessing, or feature extraction time.

| Overlap | Stylometric | N-gram | Fusion |
|---|---|---|---|
| 0% | 0.000136 | 0.00112 | 0.00122 |
| 25% | 0.000133 | 0.00122 | 0.00129 |
| 50% | 0.000131 | 0.00111 | 0.00126 |
| 75% | 0.000130 | 0.00112 | 0.00120 |
| 90% | 0.000135 | 0.00112 | 0.00127 |
| 95% | 0.000134 | 0.00115 | 0.00107 |
| 99% | 0.000129 | - | - |

**TABLE 12.** The total number of seconds per sample required to make a prediction for each feature type and overlap percentage. This includes sample generation, preprocessing, feature extraction, and classification time.

| Overlap | Stylometric | N-gram | Fusion |
|---|---|---|---|
| 0% | 0.093 | 0.175 | 0.266 |
| 25% | 0.115 | 0.219 | 0.332 |
| 50% | 0.116 | 0.202 | 0.317 |
| 75% | 0.113 | 0.174 | 0.286 |
| 90% | 0.118 | 0.142 | 0.259 |
| 95% | 0.123 | 0.145 | 0.265 |
| 99% | 0.115 | | |

report the fastest classification time per sample across all overlap percentages. On the other hand, fused features are the slowest across all overlap percentages, leaving n-grams in the middle. The most stark difference reported is found between stylometric features and fused features. In some cases such as 0%, 25%, and 50% overlap, stylometric features classify new samples almost 3 times as fast per sample, as fused features. Therefore, we can conclude that stylometric features classify new samples most efficiently.

### C. SUMMARIZING PERFORMANCE AND TIME EVALUATIONS: WHICH ONE IS THE OPTIMAL MODEL?

The results of our experiments show that the best performance metrics (accuracy, precision, recall) were achieved when maximizing the percentage of overlap and minimizing the step length of the sampling window. For instance, Table 8 shows that the best performance metrics were achieved using fused features with a window overlap of 95%. Table 13 shows the total amount of seconds it took to train each set of features for each overlap percentage. The fused features at 95% overlap which generated the best performance metrics, took 10,147.66 more seconds than stylometric features to complete the training process. Furthermore, fused features took 6,953.368 more seconds than n-grams to complete the training process. But, off-line training makes the differences in training time obsolete. Therefore, the focus should be primarily on the time it takes to classify a new sample. Table 12 shows that fused features classified new samples less efficiently per sample than stylometric features or n-grams. Again, focusing on the fused 95% overlap set, it reported to classify new samples twice as slow as the 95% overlap stylometric set, in-total reporting a difference of .142 seconds per sample. At 95% overlap, fused features were also slower than n-grams, but not as much showing the total difference

**TABLE 13.** The total amount of seconds it took to train each feature set, per overlap percentage. This includes sample generation, preprocessing, feature extraction, and fitting the model.

| Overlap | Stylometric | N-gram | Fusion |
|---|---|---|---|
| 0% | 344.23 | 941.91 | 1,273.49 |
| 25% | 447.74 | 1,218.44 | 1,571.19 |
| 50% | 664.46 | 1,853.39 | 2,207.14 |
| 75% | 1,267.61 | 2,444.02 | 3,655.57 |
| 90% | 3,282.18 | 5,879.45 | 8,224.32 |
| 95% | 6,761.18 | 9,955.47 | 16,908.84 |
| 99% | 31,623.73 | - | - |

per sample to be .12 seconds. Our opinion is that this time difference is insignificant if we consider the performance benefit of the fused features. From Table 8, we see that we achieve 14.49%, 13.78%, and 19.09% more accuracy, precision, and recall, respectively, using fused features in comparison to the stylometric features and 1.54%, 1.68%, and 2.25% more accuracy, precision, and recall, respectively, in comparison to the n-grams. Therefore, we can conclude that in cases where performance is the main priority, fused features should be used with 95% overlap between samples.

## VIII. CONCLUSION

Our study advances the state-of-the-art in several ways: traditionally, the task of mapping an author to a piece of text has been dominated by the use of n-grams and stylometric features. Our study shows that concatenating n-grams and stylometric features yields higher performance metrics than using either n-grams or stylometric features alone. We also provided detailed data which illustrated the time it took each type of features to complete each of the model development processes. Additionally, we show that by increasing the percentage of overlap found between samples in the sliding window, we can improve the performance of a model. Such that, it is almost a linear relationship where the greater the percentage of overlap is, the greater the accuracy, precision, and recall are. The findings from our experiments can greatly help in creating better authorship attribution models. The knowledge generated by our research has a high impact as identifying the original author of a piece of text remains a pivotal function in plagiarism detection, digital forensics, and identifying malicious emails and potential scams.

## REFERENCES

[1] H. Saevanee, N. Clarke, and S. Furnell, "SMS linguistic profiling authentication on mobile device," in *Proc. 5th Int. Conf. Netw. Syst. Secur.*, Sep. 2011, pp. 224–228.

[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] H. A. Bouarara, "Multi-agents machine learning (MML) system for plagiarism detection," *Int. J. Agent Technol. Syst.*, vol. 8, pp. 1–17, Jan. 2016.

[4] A. Rohwer, E. Wager, T. Young, and P. Garner, "Plagiarism in research: A survey of African medical journals," *BMJ Open*, vol. 8, Nov. 2018, Art. no. e024777.

[5] J. Schneider, A. Bernstein, J. V. Brocke, K. Damevski, and D. C. Shepherd, "Detecting plagiarism based on the creation process," *IEEE Trans. Learn. Technol.*, vol. 11, no. 3, pp. 348–361, Jul. 2018.

[6] E. E. Abdallah, A. E. Abdallah, M. Bsoul, A. F. Otoom, and E. A. Daoud, "Simplified features for email authorship identification," *Int. J. Secur. Netw.*, vol. 8, pp. 72–81, Aug. 2013.
[7] A. Abbasi, A. R. Javed, F. Iqbal, Z. Jalil, T. R. Gadekallu, and N. Kryvinska, "Authorship identification using ensemble learning," *Sci. Rep.*, vol. 12, p. 9537, Jun. 2022.
[8] S. Corbara, A. Moreo, and F. Sebastiani, "Same or different? Diff-vectors for authorship analysis," *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 1, pp. 1–36, Sep. 2023, doi: 10.1145/3609226.
[9] S. Baltes, R. Kiefer, and S. Diehl, "Attribution required: Stack overflow code snippets in GitHub projects," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 161–163.
[10] F. Ullah, J. Wang, S. Jabbar, F. Al-Turjman, and M. Alazab, "Source code authorship attribution using hybrid approach of program dependence graph and deep learning model," *IEEE Access*, vol. 7, pp. 141987–141999, 2019.
[11] V. Kalgutkar, R. Kaur, H. Gonzalez, N. Stakhanova, and A. Matyukhina, "Code authorship attribution," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–36, Feb. 2019.
[12] P. Mahbub, N. Z. Oishie, and S. M. R. Haque, "Authorship identification of source code segments written by multiple authors using stacking ensemble method," in *Proc. 22nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, 2019, pp. 1–6.
[13] H. M. G. Adorno, G. Rios, J. P. P. Durán, G. Sidorov, and G. Sierra, "Stylometry-based approach for detecting writing style changes in literary texts," *Computación y Sistemas*, vol. 22, no. 1, pp. 1–7, Mar. 2018.
[14] A. M. Robertson and P. Willett, "Applications of n-grams in textual information systems," *J. Document.*, vol. 54, no. 1, pp. 48–67, Mar. 1998.
[15] U. Sapkota, S. Bethard, M. Montes, and T. Solorio, "Not all character N-grams are created equal: A study in authorship attribution," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2015, pp. 93–102.
[16] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, Mar. 2011.
[17] G. Frantzeskou, E. Stamatatos, S. Gritzalis, and C. Chaski, "Identifying authorship by byte-level N-grams: The source code author profile (SCAP) method," *Int. J. Des. Eng.*, vol. 6, pp. 1–18, Jan. 2007.
[18] M. Koppel, J. Schler, and S. Argamon, "Computational methods in authorship attribution," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, pp. 9–26, Jan. 2009.
[19] T. Boran, M. Martinaj, and M. S. Hossain, "Authorship identification on limited samplings," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101943.
[20] M. Alsallal, R. Iqbal, V. Palade, S. Amin, and V. Chang, "An integrated approach for intrinsic plagiarism detection," *Future Gener. Comput. Syst.*, vol. 96, pp. 700–712, Dec. 2017.
[21] R. Zheng, J. Li, H.-C. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, no. 3, pp. 378–393, Feb. 2006.
[22] J. Li, R. Zheng, and H. Chen, "From fingerprint to writeprint," *Commun. ACM*, vol. 49, no. 4, pp. 76–82, Apr. 2006.
[23] 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, pp. 1–29, Apr. 2008.
[24] T. Solorio, S. Pillay, S. Raghavan, and M. M. Y. Gómez, "Modality specific meta features for authorship attribution in web forum posts," in *Proc. 5th Int. Joint Conf. Natural Lang. Process.* Chiang Mai, Thailand: Asian Federation of Natural Language Processing, Nov. 2011, pp. 156–164.
[25] S. H. H. Ding, B. C. M. Fung, F. Iqbal, and W. K. Cheung, "Learning stylometric representations for authorship analysis," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 107–121, Jan. 2019.
[26] A. Rexha, S. Klampfl, M. Kröll, and R. Kern, "Towards authorship attribution for bibliometrics using stylometric features," in *Proc. CLBib@ISSI*, 2015, pp. 44–49.
[27] M. Tschuggnall, B. Murauer, and G. Specht, "Reduce & attribute: Two-step authorship attribution for large-scale problems," in *Proc. 23rd Conf. Comput. Natural Lang. Learn. (CoNLL)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2019, pp. 951–960.
[28] M. Abuhamad, J. S. Rhim, T. AbuHmed, S. Ullah, S. Kang, and D. Nyang, "Code authorship identification using convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 95, pp. 104–115, Jun. 2019.
[29] S. T. Gupta, J. K. Sahoo, and R. K. Roul, "Authorship identification using recurrent neural networks," in *Proc. 3rd Int. Conf. Inf. Syst. Data Mining*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 133–137.
[30] Y. Sari, A. Vlachos, and M. Stevenson, "Continuous N-gram representations for authorship attribution," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 267–273.
[31] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Cambridge, MA, USA: MIT Press, 2015, pp. 649–657.
[32] P. Stefanovic, O. Kurasova, and R. Strimaitis, "The N-grams based text similarity detection approach using self-organizing maps and similarity measures," *Appl. Sci.*, vol. 9, p. 1870, May 2019.
[33] E. Lex, A. Juffinger, and M. Granitzer, "A comparison of stylometric and lexical features for web genre classification and emotion classification in blogs," in *Proc. Workshops Database Expert Syst. Appl.*, Aug. 2010, pp. 10–14.
[34] N. M. Sharon Belvisi, N. Muhammad, and F. Alonso-Fernandez, "Forensic authorship analysis of microblogging texts using N-grams and stylometric features," in *Proc. 8th Int. Workshop Biometrics Forensics (IWBF)*, Apr. 2020, pp. 1–6.
[35] L. Shang, L. Liu, W. Song, and M. Cheng, "The role of traditional features in authorship attribution," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 244–247.
[36] O. Fourkioti, S. Symeonidis, and A. Arampatzis, "Language models and fusion for authorship attribution," *Inf. Process. Manage.*, vol. 56, no. 6, 2019, Art. no. 102061.
[37] T. Boran, J. Voss, and M. S. Hossain, "Authorship categorization of public domain literature," in *Proc. IEEE 7th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2016, pp. 1–7.
[38] T. Phreeraphattanakarn and B. Kijsirikul, "Text data-augmentation using text similarity with Manhattan Siamese long short-term memory for Thai language," *J. Phys., Conf. Ser.*, vol. 1780, no. 1, Feb. 2021, Art. no. 012018.
[39] P. Shrestha, S. Sierra, F. González, 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. Stroudsburg, PA, USA: Association for Computational Linguistics, Apr. 2017, pp. 669–674.
[40] D. L. McCabe, "Cheating among college and university students: A north American perspective," *Int. J. Educ. Integrity*, vol. 1, no. 1, pp. 1–11, Nov. 2005.
[41] M. Corney, A. Anderson, G. Mohay, and O. Vel. (2001). *Identifying the Authors of Suspect Email*. [Online]. Available: https://eprints.qut.edu.au/8021/
[42] D. Boumber, Y. Zhang, and A. Mukherjee, "Experiments with convolutional neural networks for multi-label authorship attribution," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*. Paris, France: European Language Resources Association (ELRA), May 2018, pp. 2576–2581.

**AMAR SULJIC** received the B.S. degree in computer science from Southern Connecticut State University, New Haven, CT, USA, in 2021. Currently, he is a Data Analyst in marketing industry. His research interests include deep learning and authorship attribution.

**MD SHAFAEAT HOSSAIN** (Senior Member, IEEE) received the M.S. degree in computer science and the Ph.D. degree in computational analysis and modeling from Louisiana Tech University, Ruston, LA, USA, in 2012 and 2014, respectively. He is currently a Professor of computer science with Southern Connecticut State University, New Haven, CT, USA. His research interests include machine learning, multi-biometric verification, behavioral biometrics, user authentication in smartphones, and computer vision.