Asad Mahmood*, Faizan Ahmad, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar

A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X

Abstract: Stylometric authorship attribution aims to identify an anonymous or disputed document's author by examining its writing style. The development of powerful machine learning based stylometric authorship attribution methods presents a serious privacy threat for individuals such as journalists and activists who wish to publish anonymously. Researchers have proposed several authorship obfuscation approaches that try to make appropriate changes (e.g. word/phrase replacements) to evade attribution while preserving semantics. Unfortunately, existing authorship obfuscation approaches are lacking because they either require some manual effort, require significant training data, or do not work for long documents. To address these limitations, we propose a genetic algorithm based random search framework called Mutant-X which can automatically obfuscate text to successfully evade attribution while keeping the semantics of the obfuscated text similar to the original text. Specifically, Mutant-X sequentially makes changes in the text using mutation and crossover techniques while being guided by a fitness function that takes into account both attribution probability and semantic relevance. While Mutant-X requires black-box knowledge of the adversary's classifier, it does not require any additional training data and also works on documents of any length. We evaluate Mutant-X against a variety of authorship attribution methods on two different text corpora. Our results show that Mutant-X can decrease the accuracy of state-of-the-art authorship attribution methods by as much as 64% while preserving the semantics much better than existing automated authorship obfuscation approaches. While Mutant-X advances the state-of-the-art in automated authorship obfuscation, we find that it does not generalize to a stronger threat model where the adversary uses a different attribution classifier than what Mutant-X assumes. Our findings warrant the need for future research to improve the generalizability (or transferability) of automated authorship obfuscation approaches.

DOI 10.2478/popets-2019-0058

Received 2019-02-28; revised 2019-06-15; accepted 2019-06-16.

1 Introduction

Background. Authorship attribution aims at identifying an anonymous or disputed document's author by *stylometric* analysis (i.e., examining writing style). Stylometry exploits features such as word frequency and sentence length that reflect distinguishing characteristics of the text written by an individual. Such stylometric features enable authorship attribution because they tend to remain sufficiently consistent across different documents by an author but vary across different authors. A long line of research has employed stylometric analysis for authorship attribution. Classic applications of stylometric authorship attribution focused on the New Testament [18], works of Shakespeare [19], and the Federalist Papers [24]. Recent advances in machine learning have enabled notable progress in stylometric authorship attribution with impressive effectiveness [5], at scale [35], cross-domain [36], and even in open-world settings [44].

Motivation for Authorship Obfuscation. The development of powerful machine learning based stylometric authorship attribution methods creates a serious threat for privacy-conscious individuals such as journalists and activists who wish to publish anonymously. Law enforcement and intelligence agencies are actively using stylometry as part of a broad range of physical and behavioral biometric modalities for attribution [2]. Illustrating this are the recent attempts to attribute author-

***Corresponding Author: Asad Mahmood:** The University of Iowa, E-mail: amahmood1@uiowa.edu
Faizan Ahmad: Lahore University of Management Sciences and University of Virginia, E-mail: Fa7pdn@virginia.edu
Zubair Shafiq: The University of Iowa, E-mail: zubair-shafiq@uiowa.edu
Padmini Srinivasan: The University of Iowa, E-mail: padmini-srinivasan@uiowa.edu
Fareed Zaffar: Lahore University of Management Sciences, E-mail: fareed.zaffar@lums.edu.pk

ship of anonymous writings by whistleblowers [3, 10, 11] using stylometry [12]. To counter stylometric authorship attribution, researchers have developed new tools and techniques that can be used by individuals to obfuscate their writing style.

Limitations of Prior Art in Authorship Obfuscation. In a seminal work on authorship obfuscation, McDonald et al. [29] proposed Anonymouth to suggest changes that can be manually incorporated by users to anonymize their writing style. Since it can be challenging for users to manually incorporate the suggested changes, follow up improvements [8, 30] leveraged machine translation to automatically suggest alternative sentences that can be further tweaked by users. Researchers have also attempted to fully automate the obfuscation process. Recently, researchers have developed methods that apply predefined transformations (e.g. changing synonyms, certain parts of speech etc.) to automatically obfuscate text. More recently, researchers have employed Generative Adversarial Networks (GANs) to obfuscate different attributes such as age, gender, as well as authorship [43]. It has proven challenging for existing automated authorship obfuscation approaches to successfully evade machine learning based authorship attribution classifiers while preserving the semantics of the text.

Technical Challenges. Automated authorship obfuscation approaches have to carefully tread the trade-off between making sufficient changes to evade authorship attribution classifiers while making sure that these changes are appropriate so as to preserve the semantics. To this end, there are two major technical challenges that need to be addressed. First, the number of possible changes is fairly large even for medium-sized documents. Therefore, it is challenging to figure out the right set of changes that can achieve evasion without sacrificing semantics, especially for longer documents. Second, machine learning based automated obfuscation approaches generally need a substantial amount of text previously written by the author as training data to learn how to make appropriate changes.

Proposed Approach. In this paper, we present Mutant-X that addresses these challenges by leveraging a genetic algorithm based random search framework [21]. More specifically, given black-box knowledge of an authorship attribution classifier, Mutant-X sequentially makes changes in the input text using mutation and crossover techniques while being guided by a fitness function that takes into account attribution probability as well as semantics of the obfuscated text. Using this GA-based guided search approach, Mutant-X is able to quickly identify the right set of changes that can be made to successfully evade the authorship attribution classifier while preserving the semantics. Mutant-X does not require any manual effort on part of the user and works well even for long documents. Moreover, and crucially perhaps, it works without requiring any text previously written by the author for training.

Experimental Evaluation. We evaluate Mutant-X as a countermeasure to a variety of authorship attribution methods on two different text corpora. We also compare Mutant-X to three automated obfuscation approaches that have been leading performers in the annual authorship obfuscation competition at PAN-CLEF [4]. Mutant-X outperforms the baseline automated authorship obfuscation approaches in terms of evasion as well as preserving semantics. Mutant-X can decrease the accuracy of the state-of-the-art authorship attribution classifiers by as much as 64% in a 5-author setting while simultaneously preserving the obfuscated text's semantics much better than compared obfuscation approaches.

Key Contributions. We summarize our key contributions and findings as follows.

- **Automated document-level authorship obfuscation.** We propose an automated authorship obfuscation method (Mutant-X) that is based on genetic algorithms. In contrast to prior work [43], Mutant-X does not require training for an author who desires anonymity. Instead, given a black-box authorship attribution classifier that is aware of the author, Mutant-X can obfuscate a new document by that author. Mutant-X operates at the document-level as opposed to sentence-level (as is the case in some of the prior work). The advantage is that semantic consistency checks are done for the document as a whole and not on disconnected sentences.
- **Better safety and soundness than baselines.** We extensively evaluate Mutant-X in experiments with different numbers of authors and different datasets. We also use two types of authorship attribution classifiers, one is traditional machine learning and the other deep learning based. For each of the above experimental settings, we compare Mutant-X with the top three obfuscation approaches submitted to PAN [38] as baselines. The results demonstrate Mutant-X's superiority both in safety and soundness.
  With respect to safety (% drop in authorship attribution classification accuracy), Mutant-X drops accuracy from a minimum of 12% to a high of 64% (mean

= 37%). In contrast the baseline drop in accuracy ranges from 0% to 35% (mean = 14%). Mutant-X offers a safer obfuscation method while also maintaining a strong lead in soundness, i.e., METEOR scores [17]. The METEOR score range for Mutant-X is 0.48 to 0.55 (mean = 0.51). In contrast, the baseline METEOR scores range from 0.32 to 0.46 (mean = 0.38).

Limitations and Future Improvements. While Mutant-X outperforms baselines both in terms of safety and soundness, given black-box knowledge of the adversary's attribution classifier, we find that its performance drops significantly when this knowledge is imperfect. In other words, it struggles with generalizability (or transferability) to an unseen attribution classifier. As part of our current contributions, we present experiments testing Mutant-X's generalizability under nuanced definitions of imperfections in the assumed knowledge of the adversary's attribution tool. Specifically, we explore imperfections in the assumed knowledge along several dimensions (classifier technique, features, training data, etc.) with the hope that the research community takes these dimensions into consideration for building generalizable authorship obfuscation methods.

Paper Organization: The rest of this paper is organized as follows. In Section 2, we discuss the related research in authorship attribution and authorship obfuscation. In Section 3, we explain our method Mutant-X. In Section 4, we experimentally evaluate Mutant-X and analyze both quantitative and qualitative results. Section 5 concludes the paper.

2 Related Work

2.1 Authorship Attribution

There is a long line of research on authorship attribution. Earlier methods relied on basic word frequency analysis of a text corpus for authorship attribution. Later on, statistical and machine learning based methods which automatically learned patterns across many lexical and stylometric features became popular. More recently, deep learning based methods are being used for authorship attribution because they do not require manual feature engineering.

Word Frequency Analysis. The authorship of the 12 disputed Federalist Papers is a classic authorship attribution problem. Two people, Madison and Hamilton, claimed the authorship of these disputed papers. Mosteller and Wallace [34] and follow-ups by Holmes and Forsyth [24] used simple word frequency analysis to attribute the disputed papers to Madison. Word frequency analysis research has also used synonym preferences for common words to attribute authorship. Clark and Hannon [16] showed that authors' preferences for certain synonyms is useful to uncover their identities.

Machine Learning Methods. In seminal work on authorship attribution using machine learning, Abbasi and Chen [5] developed a comprehensive set of features, called "writeprints", for analyzing stylistic properties of text. They used support vector machines (SVM) with this feature set, as it outperformed neural networks and decision trees, for authorship attribution. Since writeprints feature set were computationally expensive to calculate, Brennan et al. [14] used SVMs with a variation of writeprints feature set called "writeprints static" and showed that it outperformed the standard writeprints feature set. Narayanan et al. [35] applied authorship attribution at a large-scale, with as many as 100,000 authors, and achieved reasonable accuracy. Their feature set was similar to writeprints with the addition of syntactic category pairs (frequency of every [A,B] pair where A is the parent of B in a part of speech parse tree of text). McDonald et al. [29] also showed that a compressed version of the writeprints feature set ("writprints limited") performs better than the original writeprints with SVM. They further used a simpler feature set, composed of 9 lexical features most of which are from writeprints, called "basic-9" with neural networks. Afroz et al. [6] used lexical and syntactic features such as n-gram frequency, parts-of-speech tags frequency, function words, punctuations, and special characters to train a SVM model for identifying look-alike accounts (doppelgangers) in underground forums.

Machine Learning Methods in Social Media. In the context of social media, Rajapaksha et al. [39] rephrased the authorship attribution problem as original content detection in social networks. Given a post, they identified its most likely author using Jenks optimization method with character level n-gram features. Almishari et al. [7] validated that multiple tweets from the same person could be linked together using simple features like unigrams and bigrams character frequencies and a naive Bayes classifier.

Machine Learning Methods for Plagiarism Detection. The problem of plagiarism detection is closely related to authorship attribution. AlSallal et al. [9] used common words and content words with SVM to detect plagiarism. Shahid et al. [42] used syntactic and lexical features with SVM to detect "spun" content.

Obfuscation Methods		Automated	Evasion Effectiveness	Semantic Preservation	No Training Required
McDonald et al. [29]	Manual incorporation of suggested changes	✗	✓	✓	✓*
Karadzhov et al. [26]	Target specific features	✓	✓	✗	✓
Mansoorizadeh et al. [28]	Replacement with synonyms	✓	✗	✓	✓
Keswani et al. [27]	Round trip translation	✓	✓	✗	✓
Shetty et al. [43]	Generative Adverserial Networks (GANs)	✓	✓	✗	✗
MUTANT-X (this paper)	Genetic Algorithms (GAs)	✓	✓	✓	✓*

Table 1. Comparative analysis of different authorship obfuscation methods from prior literature. ✓* indicates systems that assume some knowledge of the adversary's trained classifier.

Deep Learning Methods. A key limitation of prior machine learning methods for authorship attribution is their reliance on manually engineered feature sets such as writeprints and its variants. Recent advances in deep learning for text analysis have successfully used automatically generated "word embeddings" as features, alleviating the need for manual feature engineering. Howard and Ruder [25] used RNNs and unsupervised pre-training to outperform all the previous text classification approaches. They first trained a word embedding in an unsupervised fashion from a large collection of Wikipedia text dataset. Next, they fine tuned the model for text classification purposes and achieved state of the art results on many datasets. These results confirm the effectiveness of deep learning approaches and automatic feature extraction. In author attribution, Ruder et al. [40] used CNNs and outperformed many feature-based approaches. They tried character-level and word-level inputs as well as their combination and built several CNN architectures to achieve promising authorship attribution results on a dataset of blogs [41].

2.2 Authorship Obfuscation

Authorship obfuscation aims to modify a document such that it can evade an authorship attribution classifier while preserving the text's semantics. PAN [1], a shared task at CLEF (Conference and Labs of the Evaluation Forum), has conducted a variety of authorship analysis tasks, one of which is authorship obfuscation. Prior work on authorship obfuscation, some of which appeared at PAN, can be broadly categorized into rule based, semi-automated, or automated obfuscation methods.

Rule Based Obfuscation. Mansoorizadeh et al. [28], from PAN 2016, focused on replacing the high frequency words of an author with synonyms for obfuscation. More specifically, they replaced a word with one of its synonyms obtained from WordNet [33]. Karadzhov et al. [26], from PAN 2016, transformed stylometric features to their average values using different rules such as sentence splitting, stop words removal, and spelling correction to obfuscate text. The authors reported that splitting sentences and removal of certain words helps with obfuscation but significantly hurts the sensibility of the obfuscated text to a human reviewer. Castro et al. [15], from PAN 2017, used sentence simplification transformations such as replacing contractions with expansions and removing text in parenthesis if it does not contain any named entity to achieve obfuscation.

Semi-automated Obfuscation. JStylo has an adversarial tool called Anonymouth by McDonald et al. [29] which is used for authorship obfuscation. Anonymouth identifies distinctive features [5, 13] for an author and suggests modifications that impact them. For example, if the number of unique words count exceeds a certain limit, then Anonymouth suggests to replace unique words with less than 3 syllables with already existing words having 3 or more syllables. In a user study with 10 participants, 8 were able to successfully obfuscate.

Automated Obfuscation. Prior literature has used machine translation approaches for automated obfuscation. Keswani et al. [27], from PAN 2016, proposed using round-trip translation; translation from one language to another and back to the source language. The specific configuration that they used was English→German→French→English. The evaluation showed that the obfuscated text generated using machine translation suffers from poor readability. More recently, Shetty et al. [43] used Generative Adversarial Networks (GANs) to evade an age, sentiment, and authorship attribution classifier. While this approach worked reasonably well for age and sentiment obfuscation, it did not preserve the semantics for authorship obfuscation especially for large-sized documents [43]. Moreover, due to constraints in its design, it is unclear whether this approach can be readily adapted for multi-class (more than 2) authorship obfuscation.

2.3 Takeaway

Table 1 summarizes gaps in prior literature on authorship obfuscation. There is a paucity of automated methods for authorship obfuscation that can effectively evade attribution classifiers while preserving the semantics. Some existing methods (e.g. McDonald et al. [30]) require users to manually incorporate the suggested changes. Other automated approaches (e.g. Karadzhov et al. [26], Mansoorizadeh et al. [28]) use predefined rules. While these have the advantage that they do not require knowledge of the adversary's attribution tool, they are unable to effectively evade authorship attribution classifiers and/or cannot preserve the semantics. Some automated authorship obfuscation approaches (e.g. Shetty et al. [43]) rely on the availability of sufficient amount of training data which renders them less useful for cases where there are only a small number of documents available for an author seeking anonymity.

To address these gaps, we propose a genetic algorithm based approach for automated authorship obfuscation that assumes black-box knowledge of the adversary's attribution classifier. By black-box knowledge we mean that the system has access to the adversary's trained classifier and its input features. Given black-box knowledge, our system does not require additional training data. Also as we show later our approach outperforms leading authorship obfuscation methods (from PAN) that do not require such knowledge - in both safety (evasion effectiveness) and soundness (preservation of semantics). We obfuscate documents in the context of defeating two authorship attribution classifiers: one uses traditional machine learning with a writeprints feature set, the other is a deep learning based CNN classifier. These two were selected after extensive tests to identify the most effective authorship attribution classifiers for our datasets.

3 Proposed Approach

3.1 Intuition

The underlying principle of authorship obfuscation approaches is to make modifications in the original text such that the obfuscated text successfully evades the authorship attribution classifier. The most naive approach for modifications is random word replacements. While this naive approach can eventually evade the classifier, the obfuscated text would likely not be semantically similar to the original text. To maintain semantic similarity to the original text, we can replace words with their synonyms but this does not guarantee successful evasion (e.g., [26, 28]). We can also learn to identify suitable word replacements (e.g., [43]) in a supervised fashion but this requires sufficient training data which may not be available.

Our main intuition is to automatically replace words with their synonyms and then directly incorporate the effect of these replacements on authorship attribution probability as well as semantic relevance. This way, we can find suitable replacements to evade the authorship attribution classifier while keeping the obfuscated text semantically similar to the original. This approach does not require training to learn suitable replacements and can potentially work with any document irrespective of its size.

3.2 Mutant-X

Given an article A and authorship attribution classifier C, where C is able to correctly classify A as written by author a, the goal is to obfuscate A to A' such that C classifies A' as belonging to some other author $a' \neq a$ while keeping A and A' semantically similar. To solve this problem, we now present our genetic algorithm based authorship obfuscation method called Mutant-X. But first, we provide a brief overview of genetic algorithms and then explain how we leverage it to operationalize our intuition for automated authorship obfuscation.

Genetic algorithms, inspired by natural selection and evolution, iteratively evolve a population of individuals (potential solutions) to generate high-quality solutions in optimization and search problems. The starting point for a genetic algorithm is a population of individuals. Each individual in the population is represented by a vector of chromosomes. In every iteration, new offspring (individuals) are produced by mutating each individual in the current population. Mutation is basically an alteration strategy, which when applied to the chromosomes of an individual, alters its offspring. In addition to mutation-based alterations, a subset of individuals are also *crossed over*, which involves combining chromosomes of a pair of parent individuals when generating offspring. After performing mutation and crossover, the evolved offspring population (which we also refer to as mutants) is evaluated using a fitness function. The top performing mutants, based on their fitness scores, in the evolved population are retained for the next iteration

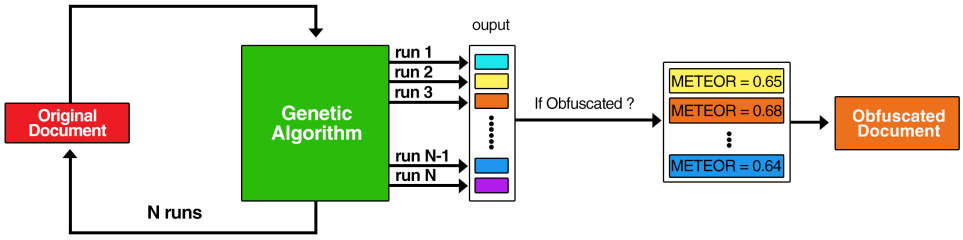

Fig. 1. Our genetic algorithm based approach (MUTANT-X) for automated authorship obfuscation.

and the remaining are discarded. The genetic algorithm terminates after reaching a stopping criteria, which is typically based on number of iterations or fitness score.

MUTANT-X is our proposed genetic algorithm based authorship obfuscation method. Figure 1 provides an overview of MUTANT-X. The population is initialized with the original article A. The chromosome representation of an individual (document) in the population is a sequence of words in the document. Mutant offsprings (altered documents) are generated by making word replacements. In iteration m, each document in the population is independently mutated to add l next generation mutants to the population. A subset of mutants are then selected for crossover with probability c, and the crossed over mutants are added to the population. Following this, a fitness function F is used to compute fitness of each mutant document in the population and only the top K (based on fitness score) mutants are retained for the next iteration. The fitness function takes into account both the detection probability of a given attribution classifier C and semantic similarity between the mutant document and the original document A. The genetic algorithm terminates when we either have an obfuscated document A' in the population. i.e., A' is no longer attributed to the original author a, or the maximum number of iterations M has been reached. Upon termination, the genetic algorithm simply returns either the obfuscated document A' for the former or the document with the highest fitness score for the latter. Owing to the stochastic nature of genetic algorithms, this process of obfuscating a document is repeated multiple times (or "runs"). Once the genetic algorithm terminates for all N runs, we select an obfuscated document among potentially multiple obfuscated versions from different runs as the one with highest semantic similarity with the original document A (based on METEOR score that is defined later in this section).

Require: original document A, maximum number of iterations M, number of mutants generated from a parent (document or previous generation mutant) l, number of word replacements in a mutation Z, crossover probability c, fitness function F, number of top individuals to select at the end of each iteration K, population set P, individual document i

P = {} ▷ initialize population set to empty
P.add(A)
for $m \in \{1, 2, ..., M\}$ **do** ▷ repeat for M iterations
  for i in P **do**
    if i is not attributed to a **then**
      return i ▷ return obfuscated document
    end if
    for $j \in \{1, 2, ..., l\}$ **do**
      P.add($mutate$(i, Z)) ▷ mutation
    end for
  end for
  P.add($crossover$(P, c)) ▷ crossover
  P ← $selectTop$(P,F,k) ▷ select top-k documents based on fitness for next iteration
end for
return $i \in$ P with highest fitness score ▷ If the document didn't obfuscate, return the document with best fitness

MUTANT-X's pseudocode for each run and its details are explained next.

3.2.1 Mutation

The goal of mutation is to alter documents by making word replacements. Words are targeted for replacement only if their part of speech is either adjective, adposition, adverb, auxiliary, conjunction, coordinating conjunction, determiner, interjection, noun, pronoun,

Default Neighbors	Sentiment Specific Neighbors
Great	Astonishing
Bad	Unbelievable
Terrific	Respectable
Decent	Solid
Nice	Commendable

Table 2. Top-5 neighbors of the word "Good" using Word2Vec word and sentiment-based word embeddings.

subordinating conjunction, or verb. Thus, for example, proper nouns are not changed.

We want to replace words with "similar" words that preserve semantics. To this end, we find word replacements using *word embeddings* such as Word2Vec [32]. Word embeddings provide mappings from words to vectors such that words similar in meaning are close to each other in the vector space. For example, the word "apple" will be closer to the word "banana" in the vector space and will be far from words such as "blog", "genes", or "kids". Word embeddings offer a sensible way for alterations in our context. At a high level, word embeddings are created by a neural network that tries to predict a word given its neighbors. More formally, given the words around the word w_i, (w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2} for window size 5), the neural network is trained to predict word w_i.

An issue with word embeddings is that sometimes words with opposite sentiment can have high similarity [45]. For example, "good" and "bad" are close in the vector space because their contexts are similar, however, they are opposite in sentiment. To avoid replacing words with opposite sentiment words, we use the approach in [46] and their code to create sentiment-specific word embeddings from the standard Word2Vec word embeddings. More specifically, given a word embedding and a sentiment lexicon, we first select the top K nearest neighbors of a certain word from the Word2Vec vector space and then reorder them based on their sentiment from the sentiment lexicon. Then the sentiment-based word embeddings are obtained by modifying the Word2Vec word embeddings using the updated order of nearest neighbors. We use this sentiment-based word embedding to make appropriate word replacements for obfuscation. Table 2 demonstrates the value of this approach by comparing the neighbors of the word "good" in Word2Vec word embeddings and in the final sentiment-based word embeddings.

For mutation (see pseudocode line 9), we replace Z words in a document. We first randomly select a word whose part of speech is in the permitted list mentioned earlier. We then look for its neighbors in the sentiment-based word embedding and select at most 5 nearest neighbors with similarity greater than 0.75. We then replace the selected word with a randomly selected neighbor from this set. Note that mutation does not change the length of the document.

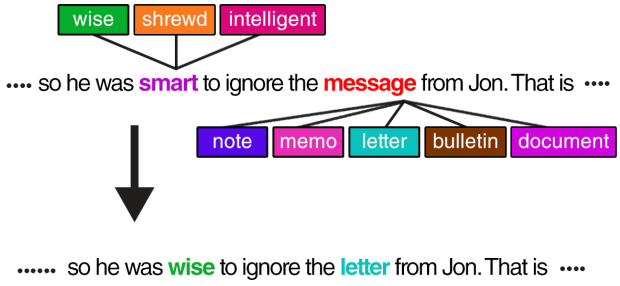

Fig. 2. Illustration of mutation using replacement by sentiment-based word embeddings

Figure 2 illustrates how mutation works with an example. We first randomly select two ($Z = 2$) words for replacement. We then look for their neighbors in the sentiment-based word embedding and find that "wise", "shrewd", and "intelligent" are neighbors of "smart". Similarly "document", "note", "letter", "memo", and "bulletin" are neighbors of "message". We then replace the word "smart" with "wise" and "message" with "letter", which are randomly selected from the sentiment-based nearest neighbor set.

3.2.2 Crossover

The goal of crossover is to produce an altered document by combining portions of two parent documents. A subset of mutated documents are randomly selected with probability c for crossover (see pseudocode line 12). We use the single-point crossover mechanism. Specifically, we select a random position in parent documents and divide both of them in two halves. We form two children documents by combining (1) the first half of the first document and the second half of second document and (2) the first half of second document and the second half of the first document.

3.2.3 Metric for Evaluation of Translation with Explicit Ordering (METEOR)

METEOR [1] [17] is a text evaluation metric based on the concept of unigram matching between two texts. Specifically, given a reference and generated text, METEOR creates a mapping between the unigrams of these two such that a single unigram in one text cannot be mapped to more than one unigram in the other. These mappings are based on the exact words, stemmed words, synonyms (extracted from WordNet) and predefined phrases from respective langauge. Counts of matched content and function words from each of these mappings are then used to calculate precision (ratio of number of unigrams mapped in the generated text to the total number of unigrams in the generated text) and recall (ratio of number of unigrams mapped in the generated text to the total number of unigrams in the reference text). Finally, the METEOR score is calculated using the following two things:

- weighted harmonic mean of precision and recall
- penalty which basically accounts for the difference in word order of matched unigrams

METEOR scores lie between 0 and 1, where the similarity progressively increases with the increase in value.

3.2.4 Fitness Function

The fitness function is used to guide the evolution of a population across iterations. For a given mutant document in the population (say x), our fitness function $F(x, A)$ has two components.

1. $C(x)$: Authorship attribution classifier confidence score for x
2. $M(x, A)$: Semantic similarity of x to the original document in terms of Metric for Evaluation of Translation with Explicit ORdering (METEOR) score [17]. METEOR is widely used for objectively measuring performance in a variety of machine translation problems because it has high correlation with subjective human judgments [17].

We define the fitness function as:

$$F(x, A) = \frac{1}{\alpha C(x) + (1 - \alpha)(1 - M(x, A)))} \quad (1)$$

Note that the fitness function tries to achieve two objectives. First, it tries to reduce the confidence of the authorship attribution classifier C. Second, it tries to ensure that the altered document is semantically similar to the original document A which we capture by computing the METEOR score between the pair of documents. The parameter α controls the relative importance assigned to the two objectives.

3.2.5 Mutant-X Deployment

Given a document that is to be obfuscated and black-box knowledge of the adversary's attribution classifier, Mutant-X performs obfuscation by altering the document using the genetic algorithms process just described. This is done N times (or over N independent "runs"). In each run, the genetic algorithm based obfuscation process alters the original document over multiple iterations (M). From the altered documents output by the different runs, we select only the obfuscated ones, i.e., those that successfully evade the given authorship attribution classifier. If multiple runs result in obfuscated documents, we give the user the version with the best METEOR score. If none of the runs result in an obfuscated document, we conclude that the input document was not obfuscated by Mutant-X within the N runs. Increasing N, which depends upon the time or computational resources available, increases the likelihood of obfuscation.

4 Evaluation

4.1 Experimental Setup

We first discuss the two publicly available datasets used for evaluation. We then briefly discuss different authorship attribution and baseline authorship obfuscation approaches. And in the end, we explain the evaluation metrics and hyperparameter selection for Mutant-X.

1 https://github.com/salaniz/pycocoevalcap

4.1.1 Data

We use two publicly available text corpora to evaluate authorship attribution and obfuscation approaches.

The **Extended-Brennan-Greenstadt** corpus [14] consists of scholarly documents (e.g., opinion papers, research papers) from 45 authors collected through the Amazon Mechanical Turk (AMT) platform. The guidelines for submissions asked the authors to not include citations, section headings, and editing notes as well as minimize the use of quotations and dialogues. The authors were also instructed to refrain from submitting samples less than 500 words. The average document size in this dataset is 492 words. The corpus contains at least 13 documents from each of the 45 authors. We refer to this corpus as the EBG dataset in the rest of the paper.

The **Blog Authorship** corpus [41] consists of more than 600 thousand blog posts from 19,320 bloggers on *blogger.com*. The corpus contains an average of 35 posts containing 7,250 words from each of the 19,320 bloggers. The average document size in this dataset is 543 words. We refer to this corpus as the BLOG dataset in the rest of the paper.

4.1.2 Authorship Attribution Approaches

Based on our literature review of prior authorship attribution approaches (see Section 2), we perform experiments by implementing the following settings (combination of feature set and classifier).

Basic-9 + FFNN [14]. Feature set used in this setting includes the following 9 features covering character-level, word-level and sentence-level features along with some readability metrics: character count (excluding whitespaces), number of unique words, lexical density (percentage of lexical words), average syllables per word, sentence count, average sentence length, Flesch-Kincaid readability metric [37], and Gunning-Fog readability metric [22]. As suggested by [14], we use the basic-9 feature set with a Feed Forward Neural Network (FFNN) whose number of layers were varied as a function of (number of features + number of target authors)/2.

Writeprints-Static [14]. This feature set includes lexical and syntactic features. Lexical features include character-level and word-level features such as total words, average word length, number of short words, total characters, percentage of digits, percentage of uppercase characters, special character occurances, letter frequency, digit frequency, character bigrams frequency, character trigrams frequency and some vocabulary richness features. Syntactic features include counts of function words (e.g., for, of), POS tags (e.g., Verb, Noun) and various punctuation (e.g., !;:). As suggested by [14], these features are used with Support Vector Machine (SVM) classifier using polynomial kernel. We also perform experiments using these features with Random Forest Classifier (RFC) using 50 decision trees.

Writeprints-Limited [29]. As compared to writeprints-static, this feature set comprises of static features as well as dynamic features based on n-gram analysis. The set of dynamic features include frequency of character bigrams and trigrams, POS tag bigrams and trigrams, bag-of-words, word bigrams and trigrams. For all the dynamic features, we only keep the values for top 50. As suggested by [29], we use these features with Support Vector Machine (SVM) classifier using polynomial kernel. Similar to the writeprints-static, we also perform experiments using these features with Random Forest Classifier (RFC) using 50 decision trees.

Word Embeddings + CNN [40]. We also use Convolutional Neural Network (CNN) classifier with word embeddings for authorship attribution [40]. More specifically, each word is mapped to a continuous-valued word vector using Word2Vec. Then each input document is represented as a concatenation of word embeddings where each word embedding corresponds to a word in original document. Using these document representations as input, we train a multi-channel CNN consisting of a static word embedding channel (word vectors trained by Word2Vec) and a non-static word embedding channel (word vectors trained initially by Word2Vec then updated during training). Training is done for 15 epochs using a batch size of 50.

We evaluate the aforementioned authorship attribution approaches for both EBG and BLOG datasets for a set of 5 and 10 authors. For both datasets, we select the authors with highest average number of characters per document. Since the EBG dataset has a varying number of documents per author (13 – 23), the training set has 12 documents per author and the remaining documents are used for testing. In BLOG dataset, every author has 100 documents, out of which we use 80 documents per author for training and the remaining 20 for testing.

Table 3 summarizes the authorship attribution accuracy of different methods for both datasets. Amongst the results for Writeprints, Writeprints-Static has the lead over Writeprints-Limited. Within Writeprints-

Authorship Attribution Method	Classifier Type	Accuracy EBG Dataset		Accuracy BLOG Dataset	
		5 Authors	10 Authors	5 Authors	10 Authors
Basic-9 + FFNN	ML	60.0%	32.6%	67.0%	54.5%
Writeprints-Limited + SVM	ML	63.3%	49.0%	76.0%	50.0%
Writeprints-Limited + RFC	ML	76.6%	55.1%	91.0%	82.0%
Writeprints-Static + SVM	ML	**90.0%**	**69.4%**	88.0%	79.5%
Writeprints-Static + RFC	ML	87.0%	**69.4%**	93.0%	**87.0%**
Word Embeddings + CNN	DL	73.0%	59.2%	**97.0%**	85.5%

Table 3. Classification accuracy of different machine learning (ML) and deep learning (DL) based authorship attribution method(s) on the EBG (5 and 10 authors) and BLOG (5 and 10 authors) datasets.

Static, Random Forest classifier has the lead over SVMs. Word embeddings with CNN has the best results for BLOG-5 and second highest for EBG-10 and BLOG-10. Thus, we decide to use both *writeprints-static + Random Forest Classifier* and *word embeddings + CNN* as the authorship attribution classifiers. For the rest of paper, we refer to *writeprints-static + Random Forest Classifier* as 'writeprintsRFC' and *word embeddings + CNN* as 'embeddingCNN'.

4.1.3 Authorship Obfuscation Approaches

As per the overview of the author obfuscation task at PAN 2018 [38], the top-3 obfuscation methods, according to their "world-ranking" metric, are presented in [15, 26, 27]. We compare MUTANT-X with these three obfuscation approaches.

Stylometric Obfuscation [26]. The stylometric obfuscation approach by Karadzhov et al. [26] involves identifying the stylometric features of an author's text and apply ad hoc transformations (e.g. splitting or merging sentences, add/remove stop words, correct spellings or add common spelling mistakes, word replacement using WordNet, etc.) on the text to make the values of the features close to the average of corpus to make them less discriminative. These features include sentence word count, punctuation to word count ratio, stop words to word count ratio, etc. To evaluate this approach, we use the code provided by the authors [26]. We refer to this approach as 'stylometricPAN16' for the rest of this paper.

Sentence Simplification [15]. Castro et al. [15] aims at obfuscating text by applying some sentence simplification transformations. This approach makes transformations like replacing contractions with expansions, removing text in parenthesis if it does not contain named entity, removing discourse markers, removing appositions (explanations for named entities), and replacing words using WordNet. To evaluate this approach, we implement it ourselves since the code was not made available. We refer to this approach as 'simplificationPAN17' for the rest of this paper.

Machine Translation [27]. The machine translation approach by Keswani et al. [27] obfuscates text by translating it from English to intermediate languages before translating it back to English. The back-and-forth translation approach changes the vocabulary as well as the structure of the original text. This approach translates the original text from English to German, German to French and then from French to English. To evaluate this approach, we implement it ourselves since the code was not made available. We refer to this approach as 'NMTPAN16' for the rest of this paper.

4.1.4 Evaluation Metrics

We evaluate the performance of obfuscation approaches in terms of the following evaluation metrics.

1. **Safety (Evasion Effectiveness)**. Obfuscation is considered successful and therefore safe when the text is no longer attributed to the original author. We measure the extent of safety by the drop in attribution accuracy (for the ground truth author) between the original and the obfuscated text.
2. **Soundness (Semantic Similarity**). Obfuscation is considered sound when the obfuscated text is semantically similar (i.e., conveys the same message) to the original text. To this end, we compute the METEOR score [17] between the original text and the obfuscated text. The METEOR score ranges is $[0, 1]$, where 1 indicates ideal score (perfect semantic relevance) and 0 indicates the opposite.

Obfuscation Method	Setting	EBG 5 Authors		EBG 10 Authors		BLOG 5 Authors		BLOG 10 Authors	
		Drop	METEOR	Drop	METEOR	Drop	METEOR	Drop	METEOR
stylometricPAN16	writeprintsRFC	7.0%	0.45	22.4%	0.45	12.0%	0.35	23.7%	0.34
simplificationPAN17	writeprintsRFC	30.3%	0.36	24.5%	0.38	7.0%	0.38	35.0%	0.38
NMTPAN16	writeprintsRFC	10.0%	0.39	22.5%	0.39	5.0%	0.32	29.1%	0.34
Mutant-X	**writeprintsRFC**	**64.0%**	**0.54**	**59.2%**	**0.55**	**24.0%**	**0.48**	**39.5%**	**0.51**
stylometricPAN16	embeddingCNN	6.6%	0.46	0.0%	0.46	10.0%	0.34	33.5%	0.38
simplificationPAN17	embeddingCNN	0.0%	0.38	2.0%	0.38	1.0%	0.40	13.5%	0.42
NMTPAN16	embeddingCNN	0.0%	0.40	6.2%	0.39	6.0%	0.32	25.5%	0.33
Mutant-X	**embeddingCNN**	**20.0%**	**0.49**	**34.7%**	**0.53**	**12.0%**	**0.52**	**44.5%**	**0.50**

Table 4. Results for different obfuscation methods on EBG and BLOG datasets using wrteprintsRFC and embeddingCNN setting. Drop indicates safety and METEOR indicates soundness. As an example, the first row indicates that the stylometricPAN16 obfuscation method with writeprintsRFC setting drops classification accuracy by 7.0% keeping the average METEOR score for obfuscated documents at 0.45.

4.1.5 Hyperparameter Selection

We conduct pilot experiments to select suitable hyperparameter values for MUTANT-X. These experiments are conducted on the training portion of the EBG 5 author dataset which is distinct from the EBG 5 author test set. This training portion consists of 60 documents (12 for each of the 5 authors). Specifically, we try a range of values for:

- number of word replacements per mutation: $Z \in \{5 \text{ words}, 10 \text{ words}, 1\% - 5\% \text{ of document length}\}$
- number of iterations: $M \in \{15, 20, 25\}$
- number of runs per document: $N \in \{10, 100\}$
- weight assigned to attribution confidence relative to METEOR score: $\alpha \in \{0.5, 0.75, 0.85\}$

Number of top individuals retained in each iteration (K) and the number of document mutants (l) are conservatively set at 5. Based on our pilot experiments on the EBG-5 dataset, we select $Z = 5\%$ of document length, $M = 25$, $N = 100$, and $\alpha = 0.75$ as providing good safety and soundness results. Note that we do not separately optimize hyperparameters for other dataset - classifier settings. Instead, we use the same hyperparameter values that are selected for the EBG-5 dataset.

4.2 Results

The results for MUTANT-X and all the obfuscation baselines are summarized in Table 4. For safety, we report the difference in classification accuracy before and after applying the obfuscation method i.e., the drop in attribution classifier's accuracy. For soundness, we report the average METEOR computed over all the successfully obfuscated documents.

Results are presented for eight distinct obfuscation experiments, where an experiment is a particular combination of classifier setting, dataset, and number of authors. As we can see from the table, MUTANT-X outperforms all baselines in all experiments both in terms of safety and soundness.

Drop in attribution accuracy is the best for MUTANT-X which is 12% to 64%, with an average of 37%. In contrast, the three baseline methods appear to struggle on safety especially with the EBG dataset combined with embeddingCNN. Drop in attribution accuracy for baselines (assessed over all baselines as a group for each obfuscation experiment) range from a low of 0% to a high of 35%, with an average of 14%.

MUTANT-X has the best soundness score, i.e., with METEOR ranging from 0.48 to 0.55 with average = 0.51. Again, in contrast, METEOR for baselines (assessed over all baselines as a group for each obfuscation experiment) ranges from 0.32 to 0.46, average = 0.38.

In summary, these results reflect MUTANT-X's superiority in terms of both safety and soundness as compared to the baselines. It is noteworthy that these baselines correspond to the top three obfuscation methods reported in PAN 2018 overview of author obfuscation task [38].

Next, we further analyze MUTANT-X by exploring the impact of the number of authors, the dataset, and the classifier setting on obfuscation.

Impact of number of authors: Intuitively we expected obfuscation to become easier as the number of authors increase since the options for target author

would increase. We find that this pattern holds with one exception (out of 4 cases). In particular it does not hold for the EBG dataset when writeprintsRFC is used.

Impact of dataset: Intuitively we expect it to be more challenging for Mutant-X to obfuscate for the BLOG dataset than for EBG as there are more documents per author in the BLOG dataset. This in turn would likely make the BLOG attribution classifier stronger and thus obfuscation more challenging. Again we see this trend holds with one exception (out of 4 cases). For BLOG-10 with embeddingCNN the drop in classification accuracy is more than for EBG-10.

Impact of classifier setting: Intuitively it should be more challenging to evade embeddingCNN as compared to writeprintsRFC since Mutant-X uses nearest-neighbors from the word embedding feature space to make replacements. This feature space could be very close to the original feature vector making the document change slightly in the feature space, thus making it harder to evade the CNN classifier. This is also seen in the results table again with one exception, the BLOG-10 dataset.

Takeaway: Our analysis of Mutant-X results across number of authors, dataset and classifier indicate that there are some trends though in each case with an exception. We suspect that interactions of these dimensions with factors such as length of text and type of text are likely perturbing the trends across settings. In future work we will need to carry out larger scale experiments across multiple datasets to disentangle and assess underlying factors.

4.3 Discussions & Limitations

Next, we study Mutant-X obfuscation to better understand its inner workings and limitations.

Impact of obfuscation on features: Figure 3 shows the writeprints-static features with the most average percentage change due to obfuscation. Specifically, we take all the documents that were obfuscated by Mutant-X and calculate the percentage of change for each writeprints-static feature. Figure 3 shows that increasing the precentage of hapax ratio (number of unique words), spaces, 'ld', 'si' and 'ra' bigrams and 'ent', 'ver', 'ter' and 'ing' trigrams, in a document helped in obfuscation. On the other hand, decreasing the percentage of '-' character helped in obfuscation. Some of the features with lowest average percentage of change (not shown in the figure) include percentage of determiners (a, an, the), punctuation and adpositions (in, to , during) and 'l' and 'w' unigrams. That is, these features were hardly changed in the obfuscated version of documents. It is to be noted that changes in these features are not targetted by Mutant-X. Mutant-X makes word replacements according to the feedback from fitness function, and these changes are a by-product of these word replacements.

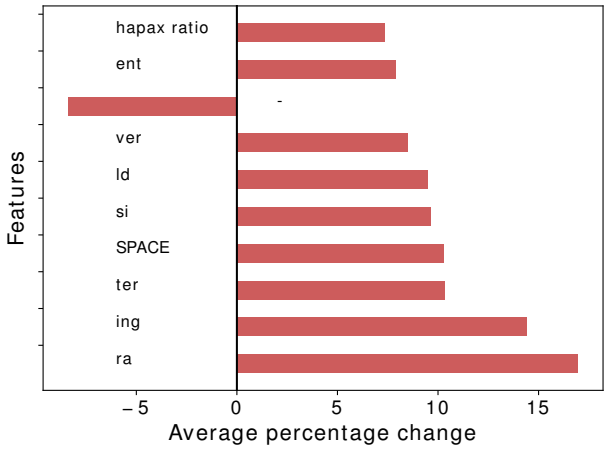

Fig. 3. Top ten writeprints-static features with respect to average percentage of change, from obfuscated documents, after applying Mutant-X on EBG dataset using 5 authors.

Understanding style transfer by Mutant-X: Style transfer refers to the manner in which the style of a document is changed with obfuscation [20, 43]. It may be that the authorship classifier is tricked into selecting a wrong author for the document. Alternatively, it may be that the style becomes transferred into some neutral point from which the classifier is unable to select any author. In Mutant-X, obfuscation style transfer is of the former kind and we investigate this further with the help of Figure 4. The figure shows the percentage of times, documents by a source author (Y-axis) become obfuscated to a target author (X-axis) using writperintsRFC as the authorship attribution classifier. We observe that authors 1, 3 and 4 prefer author 5 the most for obfuscation. Although author 4 has another strong preference of author 2 as well. Author 2 preferred author 4 the most for obfuscation and author 5 didn't obfuscate at all. This anomaly with author 5 is because there was only 1 test document for this author which did not get obfuscated.

Viewed differently, author 1 and 3 are least attractive as a target author while author 5 is the most com-

mon obfuscation target. Authors 3 and 4 never obfuscate to each other.

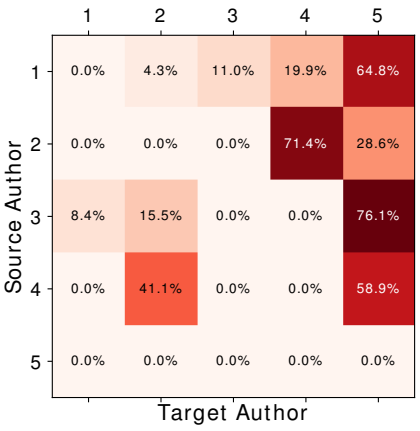

Fig. 4. Percentage of times a source author becomes obfuscated to a particular target author in all runs for EBG dataset with 5-Authors and writperintsRFC setting.

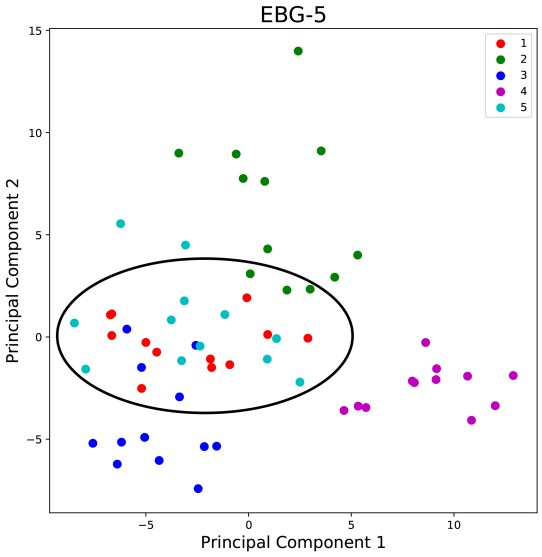

Fig. 5. Documents from the training set of EBG-5 setting represented in 2D latent space with PCA using writperint-static features.

To glean a plausible reason for why some authors are more common obfuscation targets in MUTANT-X, we wanted to analyze the distribution of documents by each author on a latent semantic space using a 'global' perspective, i.e., considering all documents from all authors jointly. Principal Component Analysis (PCA) provides this global perspective [31] so we visualized the training documents of EBG-5 in a 2D latent space using PCA. This is shown in Figure 5. The numbers in the legend correspond to author numbers in Figure 4. We observe a cluster (circled) of documents dominated by authors 5 and 1. We postulate that when a test document is randomly mutated it is more likely to hit a member of this cluster than one outside. While this explains why we see author 5 as a frequent target it does not explain why this doesn't hold for author 1. On further inspection we found that these two authors differ in terms of classification probability rank among the 5 authors i.e., author 5 most commonly was ranked 2nd and author 1 most commonly was ranked 5th. We suspect that due to this difference, author 5 was almost always preferred over author 1. We note that this is a tentative explanation and the underlying ideas will need to be tested further in future research. Since MUTANT-X is stochastic there is the possibility that a source document's style is transferred towards any author and not necessarily the closest. This can be seen with author 2 and 4 who get obfuscated to each other. Interestingly, when using the embeddingCNN attribution classifier during obfuscation a different pattern emerges: author 1 most often is the target for the other sources while this author gets obfuscated to author 3. This suggests that style transfer by MUTANT-X may be tied to the attribution classifier.

Adversary's countermeasures: Let us make the strong assumption that the adversary somehow knows that a given document has been obfuscated by MUTANT-X. Let us also assume that the adversary somehow obtains access to Figure 4 showing MUTANT-X's obfuscation patterns. The question to ask is, to what extent can the adversary identify the true author of the document? We see that the answer depends on the target author assigned to the document. When the target author is 1 or 3 (in figure 4), the adversary can determine with full confidence that the original author is 3 and 1 respectively. For target authors 2 and 4 the adversary's confidence about the original author would decrease but with an overall inclination towards author 4 and 2 respectively. Finally for target author 5, confidence regarding one particular author will be very low. It should be noted that these are strong assumptions about the adversary's knowledge of the exact transfer patterns.

Effectiveness against different attribution classifiers: It is not atypical in prior literature to assume some knowledge of the attribution classifier used by the adversary. For example, McDonald et al. [30] (Anonymouth) and Shetty et al. [43] (A4NT) respectively as-

sumed white-box knowledge and black-box knowledge of the attribution classifier used by the adversary. Recall that Mutant-X also assumes black-box knowledge of the adversary's attribution classifier. Next, we explore a stronger threat model where Mutant-X does not have *a priori* knowledge of the adversary's attribution classifier.

We evaluate Mutant-X when the adversary's attribution classifier uses different features, classification algorithm, and universe of possible authors than the attribution classifier used in Mutant-X's fitness function. Specifically, we evaluate Mutant-X when it uses writeprintsRFC in the fitness function against an adversary using the following attribution classifiers:

0. Same attribution classifier (writeprintsRFC): This attribution classifier uses the same feature set (writeprints-static) and classification algorithm (RFC).
1. Different classification algorithm (writeprintsSVM): This attribution classifier uses the same feature set (writeprints-static) but a different classification algorithm (SVM).
2. Different feature set (writeprints-Limited RFC): This attribution classifier uses a different feature set (writeprints-limited) with the same classification algorithm (RFC).
3. Different classifier and feature set (writeprints-Limited SVM): This attribution classifier uses different feature set (writeprints-limited) and classification algorithm (SVM).
4. Different classifier and feature set (embeddingCNN): This attribution classifier uses different feature set (word-level embeddings) and classification algorithm (multi-channel CNN).
5. Different authors (writeprintsRFC with different authors): This attribution classifier uses the same feature set (writeprints-static) and classification algorithm (RFC), but is trained on a different set of authors.
6. Different number of authors (writeprintsRFC with different number of authors): This attribution classifier uses the same feature set (writeprints-static) and classification algorithm (RFC), but is trained on fewer authors. Note that this scenario is advantageous for the adversary because it has has narrowed down the universe of possible authors.
7. Meta-classifier: This attribution classifier uses voting among the aforementioned 6 attribution classifiers. This captures a scenario where Mutant-X the adversary is able to leverage multiple attribution classifiers.

Adversary's Classifier	Accuracy Pre-Obfuscation	Accuracy Post-Obfuscation	Accuracy Drop
0	87%	23%	64%
1	90%	90%	0%
2	77%	63%	14%
3	63%	73%	-10%
4	73%	77%	-4%
5	100%	100%	0%
6	100%	100%	0%
7	93%	90%	3%

Table 5. Before, after and drop in attribution accuracy for EBG-5.

Table 5 shows the adversary's attribution accuracy pre- and post-obfucation by Mutant-X. The results are reported on the EBG-5 data set when Mutant-X uses writeprintsRFC (row 0). In line with the safety results reported in Section 4.2, we note that Mutant-X is able to achieves 64% drop in accuracy when it uses the same attribution classifier as the adversary. However, Mutant-X's obfuscation effectiveness significantly degrades when the adversary uses a different attribution classifier. Most notably, the adversary is able to retain the attribution accuracy of 100% when classifier 5 or 6 are used. In some cases i.e., classifier 3 and 4, the attribution accuracy actually increases. These results show that Mutant-X's effectiveness as designed right now is dependent on the black-box knowledge of the attribution classifier used by the adversary.

It is noteworthy that, even with the assumption of black-box knowledge of the adversary's attribution classifier, Mutant-X advances the state-of-the-art by fully automating obfuscation while achieving better safety and soundness results than baseline methods. As part of our future work, we plan to "generalize" Mutant-X's obfuscation by (1) continuing obfuscation even after it evades the specific attribution classifier in its fitness function and (2) using multiple attribution classifiers in its fitness function.

4.4 Qualitative Analysis

So far we have evaluated Mutant-X's obfuscation quantitatively (e.g., in terms of safety and soundness metrics). Next, we qualitatively analyze results to high-

No.	Original Sentence	Obfuscated Sentence
1	The ability to provide services is often heavily reliant on funding	The ability to give services is often heavily reliant on funding
2	Another issue is the added level of coordination that occur	Another issue is the added level of communication that occur
3	What are the ramifications of this study?	What are the ramifications of this survey?
4	If this is a fact then it would make sense...	If this is a hunch then it would make sense...
5	The fact is that it's impossible for yourself to keep all of your information completely private.	The fact is that it's unattainable for yourself to keep all of your information completely private.
6	This sort of passive monitoring of your reputation should be in everyone's online reputation management toolbox.	This sort of passive monitoring of your popularity should be in everyone's online reputation management wrench.
7	They would attend in the mornings, bringing their babies along.	They would attend in the mornings, bringing their infants along.
8	Other good news today — tried on the Measurement Shorts, and I reckon I see some progress.	Other solid news today — tried on the Measurement Shorts, and I reckon I envision some progress.
9	However, the conflict in Nepal had its own set of rules that likely don't exist in other areas.	However, the conflict in Nepal had its own set of rules that likely don't reside in other areas.
10	In 1941, a group of women in Northern Virginia formed a book club.	In 1941, a organization of women in Northern Virginia formed a book club.

Table 6. Sentences from documents showing different changes made by Mutant-X.

light different types of modifications Mutant-X makes for obfuscation.

1. **Synonym replacements.** Mutant-X often replaces the original word with one of its synonyms since it is more probable for a synonym to be the nearest neighbor in the word embedding. Some changes of this sort are (provide⟷give), (expensive⟷overpriced), (influence⟷impact), (talented⟷skillful) and so on. There was also one instance where a slang word was replaced by its correct slang synonym (lol⟷haha).
2. **Sentiment preserving changes.** Recall that we use sentiment preserving word embeddings. The merit of this strategy is seen by alterations such as (good⟷solid), (impossible⟷unattainable).
3. **POS tag preserving replacements.** The nearest neighbor of a word usually has the same POS tag as illustrated in the following examples. Noun (group⟷organization), Auxiliary verb (must⟷should), adjective (impossible⟷unattainable), Verb (run⟷walk), Pronoun (who⟷whom).
4. **Change in word form.** Sometimes the replaced word's lemma remains the same but the word form changes. For instance (refused⟷refusing), (restricted⟷restriction). Sometimes the word is changed from singular to plural e.g., (infrastructure ⟷infrastructures).

Table 6 shows some examples of different kinds of changes made by Mutant-X. We note that the changes usually make sense. For example, the modifications in sentences 1, 5, 7, 8 and 9 make perfect sense. There are a couple of instances where replacements somewhat changed the meaning, including sentence 4 (fact⟷hunch) and sentence 6 (toolbox⟷wrench). Although, the meaning change in sentence 4 is fairly subtle. Sometimes replacements ended up breaking the grammar. For instance in sentence 10, 'a group' was changed to 'a organization' instead of 'an organization'.

We further study the quality of the Mutant-X's obfuscation in terms of *smoothness*. We measure text smoothness using a word-based Neural Language Model (NLM) [43]. This NLM is trained for the EBG and BLOG datasets independently. First, to get rid of the noise, we remove sentences containing low frequency words [23] and then train the models using word sequences obtained from each sentence. We then compute smoothness by assessing the likelihood of the subsequent word as we move from left to right in a sentence. This likelihood is then averaged over the whole document. The smoothness scores are reported as the average of the log of probability estimates. 0 represents the ideal smoothness score. Lower (negative) values represent progressively worse smoothness scores.

The average smoothness of the original documents is far from perfect at -10.54. In comparison, the average smoothness of Mutant-X obfuscated documents is -11.50. The best baseline (NMTPAN16) achieves -10.28

average smoothness and the worst baseline (simplificationPAN17) achieves -12.35. We conclude that the text obfuscated using MUTANT-X and baselines have comparable smoothness. However, recall that the baselines, including NMTPAN16, suffer from poor safety because they tend to make much fewer changes. This gives baselines slight edge in smoothness but at the cost of much worse safety. MUTANT-X, by far, provides the best safety with smoothness scores that are in the mix of the methods.

5 Conclusion

We presented MUTANT-X, which is an automated authorship obfuscation method that is based on genetic algorithms. MUTANT-X sequentially makes changes in the input text using mutation and crossover techniques while being guided by a fitness function that takes into account both attribution probability and semantics of the obfuscated text. The fitness based guidance provided by MUTANT-X's genetic algorithm approach enables it to identify the right set of changes that can be made to successfully evade the authorship attribution classifier while preserving the semantics. Thus MUTANT-X, given the black-box knowledge of the target authorship attribution classifier, is readily usable for obfuscating documents of different lengths without requiring any training. The evaluation showed that MUTANT-X on average dropped the attribution accuracy by 37% while maintaining a METEOR score of 0.51.

In future, we plan to improve MUTANT-X along the following avenues. First, we plan to incorporate different types of transformations in MUTANT-X's mutation process such as bigram or phrase replacements. This can help improve MUTANT-X's evasion effectiveness against deep learning based attribution classifiers (e.g., embeddingCNN) as well as improve semantic consistency by avoiding grammatical mistakes as seen during our qualitative analysis. Second, we plan to incorporate explicit style transfer in MUTANT-X by incorporating attribution probability of target authors (in addition to that for original author) in MUTANT-X's fitness function. This would allow users to specify the target author they want to mimic which may sometimes offer a better safety option for an author seeking anonymity. Finally, we plan to improve MUTANT-X's effectiveness in a stronger threat model where it does not have black-box knowledge of the adversary's attribution classifier. Improving obfuscation effectiveness given imperfect knowledge of the adversary's attribution classifier is a worthy challenge for the research community going forward.

Acknowledgement

The authors would like to thank Muhammad Awais Jafar (Lahore University of Management Sciences) for his help during the initial stages of this research. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] PAN. https://pan.webis.de/tasks.html.

[2] Modalities – FBI. https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/biometric-center-of-excellence/modalities.

[3] I Am Part of the Resistance Inside the Trump Administration. Opinion, The New York Times. https://www.nytimes.com/2018/09/05/opinion/trump-white-house-anonymous-resistance.html, 2018.

[4] Author Obfuscation. https://pan.webis.de/clef18/pan18-web/author-obfuscation.html, 2018.

[5] A. Abbasi and H. Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, 26(2):7, 2008.

[6] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy. Doppelgänger finder: Taking stylometry to the underground. In *IEEE Symposium on Security and Privacy (IEEE S&P)*, pages 212–226. IEEE, 2014.

[7] M. Almishari, D. Kaafar, G. Tsudik, and E. Oguz. Stylometric linkability of tweets. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES 2014)*, pages 205–208. ACM, 2014.

[8] M. Almishari, E. Oguz, and G. Tsudik. Fighting Authorship Linkability with Crowdsourcing. In *ACM Conference on Online Social Networks (COSN)*, 2014.

[9] M. AlSallal, R. Iqbal, S. Amin, A. James, and V. Palade. An Integrated Machine Learning Approach for Extrinsic Plagiarism Detection. In *9th International Conference on Developments in eSystems Engineering*. IEEE, 2016.

[10] Anonymous. I'm an Amazon Employee. My Company Shouldn't Sell Facial Recognition Tech to Police. https://medium.com/s/powertrip/im-an-amazon-employee-my-company-shouldn-t-sell-facial-recognition-tech-to-police-36b5fde934ac, 2018.

[11] Anonymous. An Open Letter to Microsoft: Don't Bid on the US Military's Project JEDI. https://medium.com/s/story/an-open-letter-to-microsoft-dont-bid-on-the-us-military-s-project-jedi-7279338b7132, 2018.

[12] S. Borenstein. Close Look at Word Choice Could ID Anonymous NYT Columnist: Word Detectives. https:

//www.nbcchicago.com/news/politics/Science-May-Help-Identify-Opinion-Columnist-492649561.html, 2018.

[13] M. Brennan and R. Greenstadt. Practical Attacks Against Authorship Recognition Techniques. In *Proceedings of the Twenty-First Conference on Innovative Applications of Artificial Intelligence*, 2009.

[14] M. Brennan, S. Afroz, and R. Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. In *ACM Transactions on Information and System Security (TISSEC)*, volume 15, 2012.

[15] D. Castro-Castro, R. O. Bueno, and R. Munoz. Author Masking by Sentence Transformation. In *Notebook for PAN at CLEF*, 2017.

[16] J. H. Clark and C. J. Hannon. An Algorithm for Identifying Authors Using Synonyms. In *Eighth Mexican International Conference on Current Trends in Computer Science (ENC 2007)*, pages 99–104. IEEE, 2007.

[17] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. 10.3115/v1/W14-3348. URL https://www.aclweb.org/anthology/W14-3348.

[18] M. Ebrahimpour, T. J. Putnins, M. J. Berryman, A. Allison, B. W.-H. Ng, and D. Abbott. Automated Authorship Attribution Using Advanced Signal Classification Techniques. *PLOS ONE*, 2013. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0054998.

[19] O. Ehmoda and E. Charniak. Statistical Stylometrics and the Marlowe-Shakespeare Authorship Debate. Master's thesis, Department of Cognitive, Linguistic & Psychological Sciences, Brown University, 2012.

[20] C. Emmery, E. Manjavacas, and G. Chrupala. Style Obfuscation by Invariance. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[21] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[22] R. Gunning. The Fog Index After Twenty Years. *International Journal of Business Communication*, 1969.

[23] J. Guo, S. Lu, H. Cai, W. Zhang, and Y. Yu. Long text generation via adversarial training with leaked information. In *AAAI*, 2018.

[24] D. I. Holmes and R. S. Forsyth. *The Federalist revisited: New directions in authorship attribution*. PhD thesis, Department of Mathematical Sciences, University of the West of England, Bristol, UK, 1995.

[25] J. Howard and S. Ruder. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume abs/1801.06146, 2018.

[26] G. Karadzhov, T. Mihaylova, Y. Kiprov, G. Georgiev, I. Koychev, and P. Nakov. The case for being average: A mediocrity approach to style masking and author obfuscation. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 173–185. Springer, 2017.

[27] Y. Keswani, H. Trivedi, P. Mehta, and P. Majumder. Author Masking through Translation. In *Notebook for PAN at CLEF 2016*, pages 890–894, 2016.

[28] M. Mansoorizadeh, T. Rahgooy, M. Aminiyan, and M. Eskandari. Author obfuscation using WordNet and language models. In *Notebook for PAN at CLEF 2016*, 2016.

[29] A. W. McDonald, S. Afroz, A. Caliskan, A. Stolerman, and R. Greenstadt. Use fewer instances of the letter 'i': Toward writing style anonymization. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 299–318. Springer, 2012.

[30] A. W. McDonald, J. Ulman, M. Barrowclift, and R. Greenstadt. Anonymouth Revamped: Getting Closer to Stylometric Anonymity. In *PETools: Workshop on Privacy Enhancing Tools*, volume 20, 2013.

[31] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, Feb. 2018.

[32] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[33] G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[34] F. Mosteller and D. Wallace. *Inference and disputed authorship: The Federalist*. 1964.

[35] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song. On the Feasibility of Internet-Scale Author Identification. In *IEEE Symposium on Security and Privacy (SP)*, pages 300–314. IEEE, 2012.

[36] R. Overdorf and R. Greenstadt. Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution. 2016.

[37] E. Pitler and A. Nenkova. Revisiting Readability: A Unified Framework for Predicting Text Quality. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[38] S. Potthast and S. Hagen. Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety. In *Notebook for PAN at CLEF 2018*, 2018.

[39] P. Rajapaksha, R. Farahbakhsh, and N. Crespi. Identifying Content Originator in Social Networks. In *IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[40] S. Ruder, P. Ghaffari, and J. G. Breslin. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv:1609.06686*, 2016. URL https://arxiv.org/abs/1609.06686.

[41] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

[42] U. Shahid, S. Farooqi, R. Ahmad, Z. Shafiq, P. Srinivasan, and F. Zaffar. Accurate detection of automatically spun content via stylometric analysis. In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 425–434. IEEE, 2017.

[43] R. Shetty, B. Schiele, and M. Fritz. A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation. In *USENIX Security Symposium*, 2018.

[44] A. Stolerman, R. Overdorf, S. Afroz, and R. Greenstadt. Classify, but verify: Breaking the closed-world assumption in stylometric authorship attribution. In *IFIP Working Group*, volume 11, page 64, 2013.

[45] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565, 2014.

[46] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang. Refining word embeddings using intensity scores for sentiment analysis. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.